# PC-98XL/XA

# テクニカルデータブック

アスキー出版局　テクライト編

PC-98XL
PC-98XA

アスキー出版局

# PC-98XL/XA
# テクニカルデータブック

アスキー出版局 テクライト編

PC-98XL
PC-98XA

アスキー出版局

## はじめに

1985年，CPUに80286を搭載し，解像度を高めてディスプレイの機能を強化し，PC-98XAはPC-9800シリーズの最上位機種として発売されました．その後1987年には，PC-9800シリーズと互換性のあるノーマルモード，PC-98XAと互換性のあるハイレゾリューションモードの2つのモードを持ち，グラフィック処理をさらに高速化したマシンとしてPC-98XLが発売されました．

PC-98XAの発売以来，処理速度およびグラフィックのパフォーマンスに注目を集め，CAD・CAMなどのプロフェッショナルユースの分野では，XLおよびXAのための様々なソフトウェアが発売されるようになってきました．

本書は，PC-98XL/XAパーソナルコンピュータの機能により高度に活用するために，ハードウェア，ソフトウェアの両面におけるデータを網羅したものです．

ソフトから見たハードウェアの動作を理解し，基本入出力プログラムを活用することは，より効率的なプログラム開発につながります．そしてハードウェアの動作の理解は，特殊な周辺装置の接続などを可能にし，PC-98XL/XAを応用した，ユーザーの目的にかなったシステムの構築に役立つものとなるでしょう．

なお本書は，限られたページの中に膨大なデータを記載するために，解説書ではなくデータブックとして作られました．そのため，本書を利用する場合には，ハードウェア，ソフトウェアについて，ある程度以上の知識を必要とします．特にハードウェアにおいては，使用しているデバイスの一般的な解説などは行っていません．必要な場合には，各デバイスのデータシートなどを参考にしてください．

本書の内容については，万全を期していますが，万一不審な点や誤り，記載もれなどがある場合には御容赦願います．また本書に記載してある事項は，特別な条件の下では正確に機能しない場合があるかも知れません．このような場合には責任が負えませんので，使用される場合には，前もって充分な評価を行ってください．

## 本書を読むに当って

### ●PC-98XLのノーマルモードについて

PC-98XLは，動作モードとしてノーマルモード，ハイレゾリューションモードの2つがある．そしてXLのハードウェアは，動作モードによって異なる状態になる．そのため本書では，XLのハードウェアに関し，ノーマルモード，ハイレゾリュションモードの双方のデータを記載してある．

ただし，PC-98XLのソフトウェア（基本入出力プログラム）に関しては，ノーマルモード時の動作が，PC-9800シリーズの各機種（PC-9801VM等）の動作と全く同等となるため，本書第3部の基本入出力プログラムの項では，PC-98XAおよびPC-98XLのハイレゾリューションモード時の動作を中心に記述し，PC-98XLのノーマルモードの動作は，PC-9800シリーズと異なる事項のみを記載した．

PC-98XLのノーマルモード時の動作については，特に記述がないかぎり，「PC-9800シリーズテクニカルデータブック」の解説を参照されたい．

### ●表記について

本書中でPC-98XLとPC-98XAの双方に共通の事項を記述する場合，NECの表記に従い「PC-98XL/XAという表記を用いた．ただし，PC-98XLのハイレゾリューションモード時とPC-98XAだけに共通し，PC-98XLのノーマルモード時に適用されない事項を記載する場合には，可読性を高めるために「PC-98XA/XLハイレゾリューションモード時」という表記を用いた．

### ●本文中のデータについて

本文中で「未使用：予約」となっている領域は，将来使用する予定があり，システム側で予約していることを表している．また，ビット情報の中で「不定」または「Don't Care」として「×」で表されているものに関しては，現在は使用されていないが，将来に渡って保証されるものではない．従って，これらの領域やビットを利用および参照するようなプログラムを作成してはならない．

●サンプルリストの実行環境

本書第3部に記載してあるサンプルリストは，以下の環境で動作確認を行ったものであり，これ以外の環境での正常動作を保証するものではない．

サンプルプログラムは，BIOS を利用してアプリケーションプログラムを開発するうえで参考とするための例であり，実用プログラムではないことに注意されたい．

ハードウェア

・PC-98XL model 1， 2， 4

・PC-98XA model 11，21，31

ソフトウェア

・日本語 MS-DOS（Ver2.0）　PS98-XL121-HMW

・日本語 MS-DOS（Ver3.1）　PS98-XL125-HMW

・日本語 MS-DOS（Ver3.1）用マクロアセンブラ　PS98-333-HMW

Microsoft Macro Assembler ver. 3.00

Microsoft Object Linker ver. 3.00

## 80286 プログラミング上の注意

PC-98XL／XA において 80286 のプログラムを作成する場合，μPD70116 および 8086 の場合と比較して以下の点の動作が異なるため，注意が必要となる．

・命令クロック数

80286 は，ほとんどの命令が μPD70116（8086）よりも少ないクロック数で実行される．従ってプログラムの実行時間によりタイミングを合わせているプログラムは変更する必要がある．

・DIV 命令の割り込み

80286 で例外発生により割り込みが起こった場合には，待避されている CS : IP は例外を発生した命令を指すのに対し，μPD70116（8086）で除算エラー割り込みが起きた場合，CS : IP は次の命令を指している．

・CPU の差異

80286 は μPD70116 の上位互換性を持っていないため，μPD70116 固有の命令は 80286 では実行できない．

μPD70116 の POP CS や MOV CS の命令は，80286 においては無効命令コードの例外（割り込み 6）を発生したり，LIDT のような保護モードの動作となってしまう．また POP MEM または PUSH MEM の第 2 バイトのビット 3～5 に未定義なパターンを使用した場合，80286 では一般保護の例外（割り込み 13）を発生する．

・PUSH SP 命令

80286 の PUSH SP 命令では，μPD70116（8086）とは異なる値をプッシュする．プッシュする値そのものが意味をもつ場合には，μPD70116 の PUSH SP と等価にするため，80286 の PUSH SP 命令を次の 3 命令に置き換える必要がある．

```
PUSH    BP
MOV     BP, SP
XCHG    BP, [BP]
```

・シフト，ローテイト

80286 において，31 ビットを超えるシフト／ローテイトカウントは，下位 5 ビットを残してマスクされる．

・IDIV の商

80286 では IDIV の命令の商として最大の負数(80H，8000H)を生成することが出来るが，μPD70116 では除算エラー割り込みが発生する．

・ステータスレジスタ

80286 ではステータスレジスタのビット数が追加されているため，ステータスレジスタの内容をテストしているプログラムにおいて，追加されたビットがマスクされていない場合に誤動作となることがある．

・NDP の割り込み

NDP からの割り込み信号は，μPD70116 の場合には 8259 を介して CPU に接続されており，8259 に対してコマンドを送り，割り込みをマスクしたりベクタアドレスを変更することができるが，80286 の場合には 80287 から直接信号が接続されているため，8259 による制御は不可能である．

・プリフィックス

μPD70116 では 1 命令の語長に制限が無いが，80286 では 1 命令が 10 バイトを越えると，無効命令コードの例外（割り込み 6）が発生する．そのためプリフィックスの重複は注意を要する．

μPD70116 ではロックプリフィックスはいかなる命令にも使用できたが，80286 では，XCHG，MOV，MOVS，INS，OUTS 命令のみで使用でき，その他の命令に使用した場合には無視される．

# 目　次

# 第2部　ハードウェア

# 第３部　基本入出力プログラム

## 第4部 外部インターフェイス仕様

## 第4部 外部インターフェイス仕様

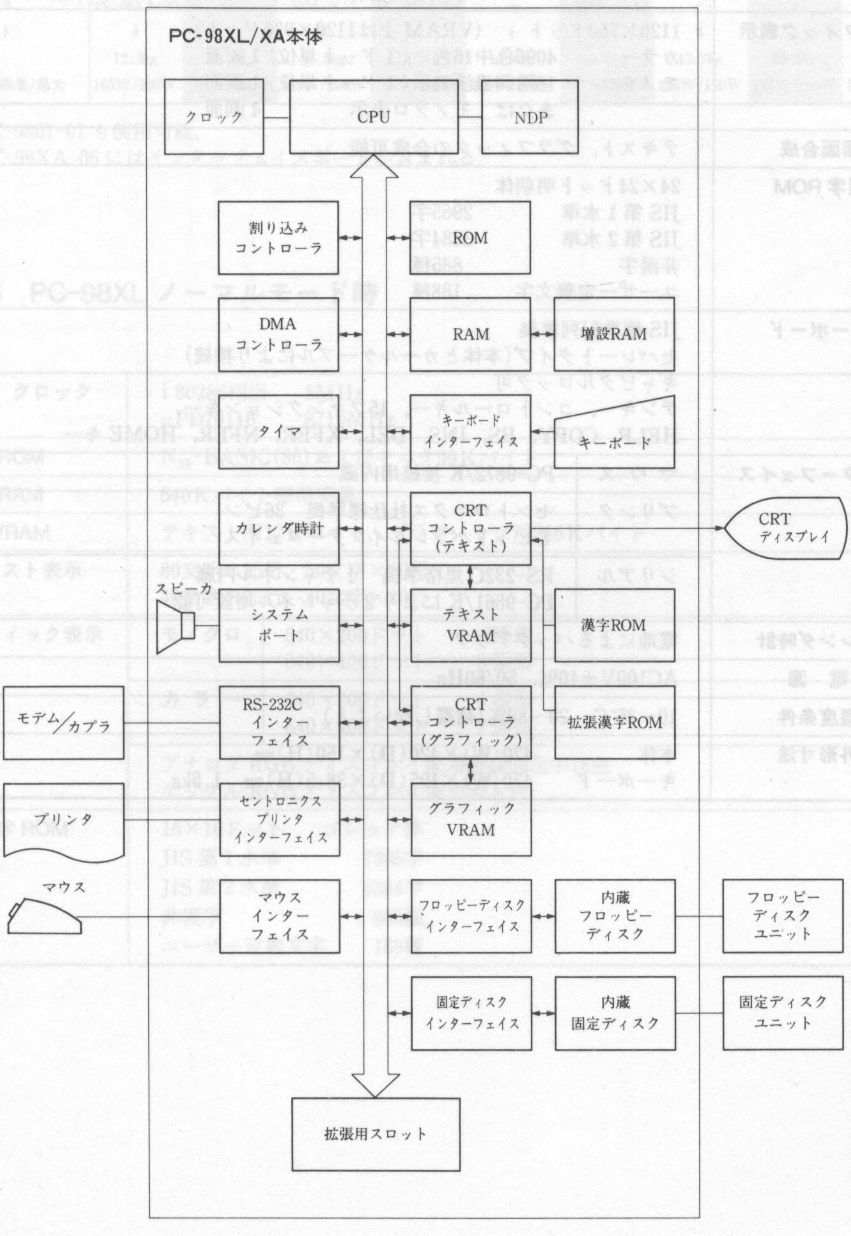
# 第1部

# PC-98XL/XAシステム構造

# 第1章

# システム概要

## 1.1 システムブロックダイアグラム



# 1.2　PC-98XL/XA 仕様一覧

## 1.2.1　共通仕様(PC-98XA/XL ハイレゾリューションモード時)

| | |
|---|---|
| VRAM | テキスト　12Kバイト<br>グラフィック　512Kバイト |
| テキスト表示 | ANK　80文字×25行<br>日本語　40文字×25行<br>アトリビュート　リバース，ブリンク，シークレット<br>カラー8色(キャラクタ単位) |
| グラフィック表示 | 1120×750ドット　(VRAM上は1120×936ドット)<br>カラー　4096色中16色　(1ドット単位)1画面<br>モノクロ　16階調濃淡表示(1ドット単位)1画面<br>または　モノクロ表示　4画面 |
| 画面合成 | テキスト，グラフィックの合成可能 |
| 漢字ROM | 24×24ドット明朝体<br>JIS第1水準　2965字<br>JIS第2水準　3384字<br>非漢字　885種<br>ユーザー定義文字　188種 |
| キーボード | JIS標準配列準拠<br>セパレートタイプ(本体とカールケーブルにより接続)<br>キャピタルロック可<br>テンキー，コントロールキー，15ファンクションキー，<br>HELP，COPY，BS，INS，DEL，XFER，NFER，HOMEキー |
| インターフェイス | マウス　PC-9872/K接続用内蔵<br>プリンタ　セントロニクス社仕様準拠　36ピン<br>8ビットパラレルインターフェイス<br>シリアル　RS-232C規格準拠　1チャンネル内蔵<br>PC-9861/Kにより2チャンネル増設可能 |
| カレンダ時計 | 電池によるバックアップ |
| 電　源 | AC100V±10%，50/60Hz |
| 湿度条件 | 10～35°C，20～80%(結露しないこと) |
| 外形寸法 | 本体　470(W)×420(D)×150(H)㎜<br>キーボード　470(W)×195(D)×38.5(H)㎜　1.5kg |

## 1.2.2 機種別仕様(本体内仕様)

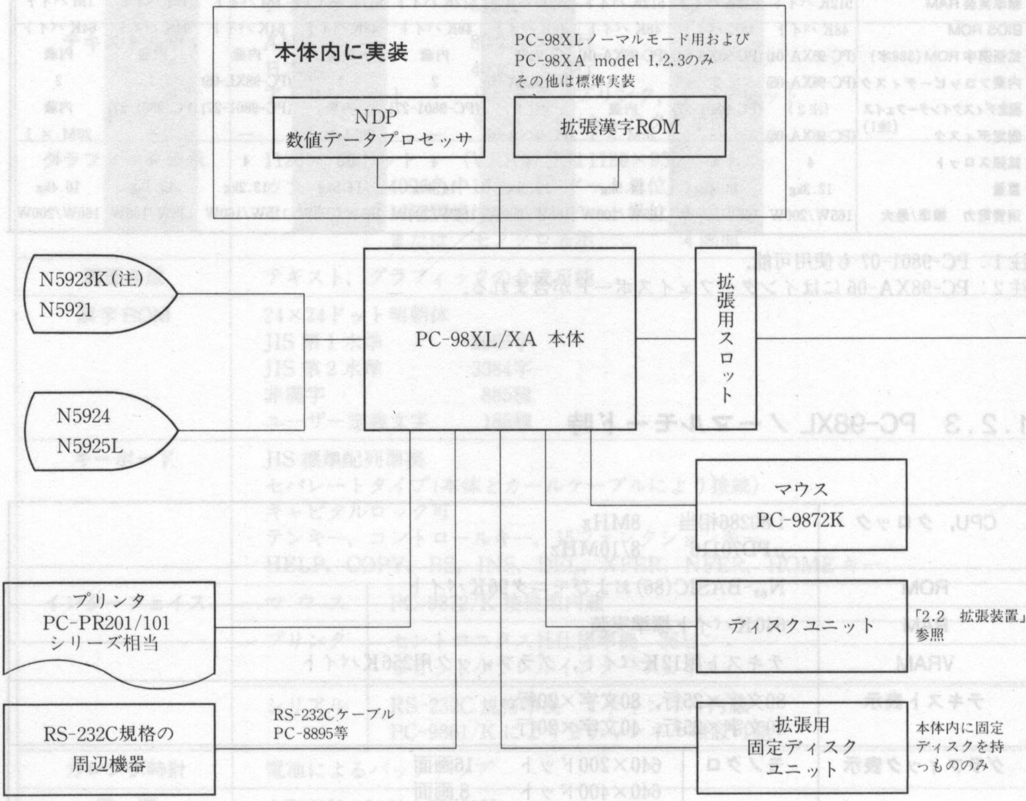
| | PC-98XA model 1 | PC-98XA model 2 | PC-98XA model 3 | PC-98XA model 11 | PC-98XA model 21 | PC-98XA model 31 | PC-98XL model 1 | PC-98XL model 2 | PC-98XL model 4 |
|---|---|---|---|---|---|---|---|---|---|
| CPU | i 80286相当 | i 80286相当 | i 80286相当 | i 80286相当 | i 80286相当 | i 80286相当 | i 80286相当 | i 80286相当 | i 80286相当 |
| クロック | 8MHz | 8MHz | 8MHz | 8MHz | 8MHz | 8MHz | 8MHz/10MHz | 8MHz/10MHz | 8MHz/10MHz |
| 標準実装RAM | 512Kバイト | 512Kバイト | 512Kバイト | 512Kバイト | 512Kバイト | 512Kバイト | 1Mバイト | 1Mバイト | 1Mバイト |
| BIOS ROM | 48Kバイト | 48Kバイト | 48Kバイト | 48Kバイト | 48Kバイト | 48Kバイト | 64Kバイト | 64Kバイト | 64Kバイト |
| 拡張漢字ROM(388字) | (PC-98XA-04) | (PC-98XA-04) | (PC-98XA-04) | 内蔵 | 内蔵 | 内蔵 | 内蔵 | 内蔵 | 内蔵 |
| 内蔵フロッピーディスク | (PC-98XA-05) | 2 | 1 | (PC-98XA-05) | 2 | 1 | (PC-98XL-05) | 2 | 2 |
| 固定ディスクインターフェイス(注1) | (注2) | (PC-9801-27) | 内蔵 | (注2) | (PC-9801-27) | 内蔵 | (PC-9801-27) | (PC-9801-27) | 内蔵 |
| 固定ディスク | (PC-98XA-06) | — | 20M × 1 | (PC-98XA-06) | — | 20M × 1 | — | — | 20M × 1 |
| 拡張スロット | 4 | 4 | 3 | 4 | 4 | 3 | 4 | 4 | 4 |
| 重量 | 12.3kg | 14.4kg | 14.5kg | 12.3kg | 14.4kg | 14.5kg | 13.2kg | 15.1kg | 16.4kg |
| 消費電力 標準/最大 | 165W/200W | 165W/200W | 165W/200W | 165W/200W | 165W/200W | 165W/200W | 115W/160W | 135W/160W | 160W/200W |

注1：PC-9801-07も使用可能.
注2：PC-98XA-06にはインターフェイスボードが含まれる.

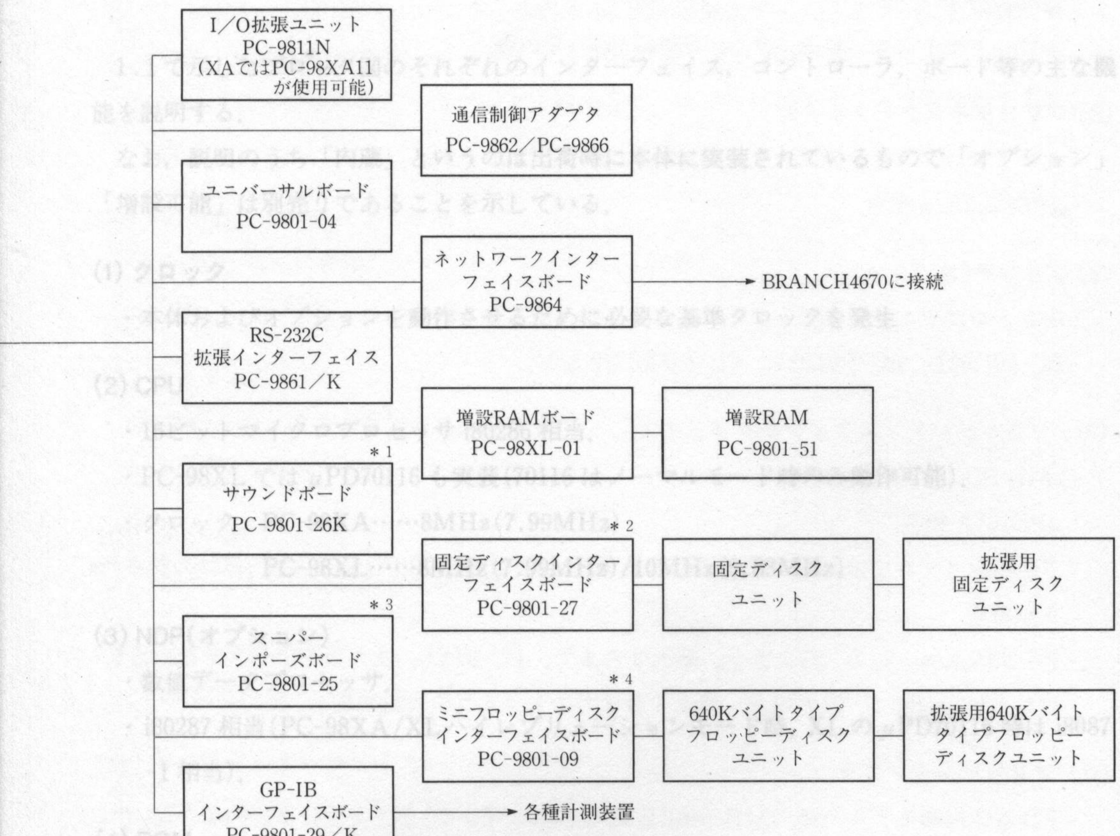
## 1.2.3 PC-98XLノーマルモード時

<table>
<tr><td>CPU，クロック</td><td colspan="3">i 80286相当　8MHz<br>μPD70116　8/10MHz</td></tr>
<tr><td>ROM</td><td colspan="3">N88-BASIC(86)およびモニタ96Kバイト</td></tr>
<tr><td>RAM</td><td colspan="3">640Kバイト標準実装</td></tr>
<tr><td>VRAM</td><td colspan="3">テキスト用12Kバイト，グラフィック用256Kバイト</td></tr>
<tr><td>テキスト表示</td><td colspan="3">80文字×25行，80文字×20行，<br>40文字×25行，40文字×20行</td></tr>
<tr><td rowspan="3">グラフィック表示</td><td>モノクロ</td><td>640×200ドット<br>640×400ドット</td><td>16画面<br>8画面</td></tr>
<tr><td>カラー</td><td>640×200ドット<br>640×400ドット</td><td>4画面<br>2画面</td></tr>
<tr><td colspan="3">アナログRGBディスプレイ使用時4096色中16色<br>デジタルRGBディスプレイ使用時8色</td></tr>
<tr><td>漢字ROM</td><td colspan="3">16×16ドット　ゴシック体<br>JIS第1水準　2965字<br>JIS第2水準　3384字<br>非漢字　885種<br>ユーザー定義文字　188種</td></tr>
</table>

# 1.3　システム接続図




注：N5923K，N5921は，PC-98XLノーマルモード時は使用不可




＊1：XLでのみ使用可能．
＊2：固定ディスク内蔵型では，同等品が標準実装されている．
＊3：XLノーマルモード時のみ使用可能．
＊4：クロック8MHz時のみ使用可能．
＊5：ROM搭載ボードをXAで使用する場合は，ROM Killの必要あり．

## 1.3 システム接続図

# 第2章

# ハードウェア概説

1.1で示したブロック図のそれぞれのインターフェイス，コントローラ，ボード等の主な機能を説明する．

なお，説明のうち「内蔵」というのは出荷時に本体に実装されているもので「オプション」「増設可能」は別売りであることを示している．

### (1) クロック

・本体およびオプションを動作させるために必要な基準クロックを発生

### (2) CPU

・16ビットマイクロプロセッサi80286相当．

・PC-98XLでは$\mu$PD70116も実装(70116はノーマルモード時のみ動作可能)．

・クロック　PC-98XA……8MHz(7.99MHz)

　　　　　　PC-98XL……8MHz(7.99MHz)/10MHz(9.83MHz)

### (3) NDP(オプション)

・数値データプロセッサ．

・i80287相当(PC-98XA/XLハイレゾリューションモード時，XLの$\mu$PD70116時はi8087-1相当)．

### (4) ROM

・本体基本回路のテストプログラム，ローダプログラムおよび，本体を構成する各ブロックを制御するプログラムを内蔵．

### (5) RAM

・標準内蔵512Kバイト(PC-98XA)

　　　　　1Mバイト(PC-98XL)

・増設可能容量は，256Kバイト単位に最大7.5Mバイトまで(XLは1枚目増設用としてPC-98XL-01(512KB)を使用し，その後は1MB単位に増設)．ただし，本体内の最大増設容量は，3.5Mバイト(固定ディスク内蔵モデル)または4.5Mバイト(XL model 4は4.5Mバイトまで)．

・PC-98XLノーマルモードの$\mu$PD70116使用時は，640Kバイトのみ使用可能．

(6) キーボードインターフェイス

・キーボードとのデータ転送を制御.

・μPD8251A 相当.

(7) キーボード

・107個のキーを持つ.

・4ビットマイクロプロセッサ(μPD8048 相当)を内蔵.

(8) プリンタインターフェイス

・セントロニクス社仕様インターフェイス(36ピンコネクタ).

・μPD8255A-5 相当.

(9) プリンタ(オプション)

・セントロニクス社仕様インターフェイスを持ったプリンタ.

・PC-PR101L, PC-PR201V, PC-PR201H2 など.

(10) フロッピーディスクインターフェイス

・1MB FD ユニットを制御.

・μPD765A 相当.

・FD 内蔵モデルでは, 1MB FD(5インチ)媒体および 640KB FD(5インチ)媒体の読み/書きが可能.

・PC-98XA model 3, 31の場合, 最大3台まで, PC-98XL, または PC-98XA model 1, 11および model 2, 21の場合, 最大4台までのドライブを制御可能.

(11) フロッピーディスク

・FD 本体内蔵モデルの時, 1MB/640KB 両用 FD(5インチ)が1または2台内蔵され, さらに, 増設用として, PC-9881K/N, PC-8882, PC-9831-MW, PC-9832-MW 等の 1MB フロッピーディスクユニットのいずれかを1台接続可能.

・本体に内蔵しないモデルの時, 1, 2台目用として PC-9881K/N, PC-9831-MW を, さらに増設用として PC-9881N, PC-9832-MW 等を接続可能.

また, 増設用ドライブユニットにより, 本体購入後, 2台内蔵モデルに変更することが可能(PC-98XA model 1, 11の場合 PC-98XA-05, PC-98XL model 1は PC-98XL-05 を使用).

(12) 固定ディスクインターフェイス

・固定ディスクを制御.

・μPD7261 相当(本体内蔵モデル).

・HD 本体内蔵モデル以外の場合, PC-9801-27 固定ディスクインターフェイスボードを使用(拡張スロットに挿入).

(13) 固定ディスク

・HD本体内蔵モデルの時，20Mバイトタイプを内蔵し，さらに，1台を増設可能(PC-98H52Nなど).

・HDを内蔵しないモデルの時，1台目用としてPC-98H33N/51Nなどを接続，さらに2台目用としてPC-98H34N/52Nなどが使用可能.

・PC-98XA model 1，11の時，PC-98XA-06増設用固定ディスクユニットを増設することにより，本体購入後，20Mバイトタイプを1台内蔵することが可能．また，PC-9801-27　固定ディスクインターフェイスボードを接続して，本体外に固定ディスクを増設することも可能.

(14) RS-232Cインターフェイス

・RS-232C規格に準拠したインターフェイス(1回線).

・μPD8251AF相当.

(15) システムポート

・本体内部のシステム情報の入出力に使用.

・μPD8255A相当.

(16) スピーカ

・タイマにより周波数の変更が可能.

(17) ディップスイッチ

・本体前面の窓の中にあるスイッチ.

(18) マウスインターフェイス

・マウスを制御.

・PC-98XA――μPD8255A相当

・PC-98XL――専用LSI

(19) マウス(オプション)

・PC-9872/Kを使用.

(20) CRTコントローラ(テキスト)

・テキスト表示を制御.

・μPD7220A相当.

(21) CRTコントローラ(グラフィック)

・グラフィック表示を制御.

・μPD7220A相当.

(22) テキスト VRAM

・テキスト画面に表示される文字の，文字コードおよびアトリビュート情報を記憶するメモリ．

・12K バイト．

(23) グラフィック VRAM

・ドット単位のグラフィックデータを記憶するメモリ．

・512K バイト(128K バイト×4プレーン)

(24) キャラクタジェネレータ

・ANK(英数カナ)文字 …………………… 244文字(特殊文字含む)

・JIS 第Ⅰ水準漢字 ………………………… 2965文字

・JIS 第Ⅱ水準漢字 ………………………… 3384文字

・非漢字・ケイ線文字 ……………………… 611文字

・ユーザー定義文字 ………………………… 188文字

・半角文字 …………………………………… 281文字

・拡張漢字 ROM(PC-98XA model 1，2，3のみオプション)．

JIS にない漢字用文字ジェネレータ … 388文字

(25) CRT

・PC-98XL では動作モードにより分解能が異なる．

a)　ハイレゾリューションモード

1120×750ドット

N5921，N5923K 等が使用可．

b)　ノーマルモード

640×400ドット

N5913，PC-KD854 等が使用可．

・PC-98XA はハイレゾリューションモードに相当する分解能のみ．

・両モードで使用するためには,N5924 等のマルチスキャンタイプのモニタが必要．

(26) 割り込みコントローラ

・μPD8259A 相当2個．

(27) DMA コントローラ

・メモリと周辺装置の高速データ転送制御．

・μPD8237A 相当．

・チャネル数4，データ転送幅8または16ビット，アドレス24ビット．

(28) **タイマ**

・RS-232C インターフェイス，スピーカ等の制御.

・μPD8253 相当.

(29) **カレンダ時計**

・時，分，秒，月，日，曜日を持つカレンダ時計.

・μPD4990A 相当(PC-98XA は μPD1990A 相当).

・大，小の月の自動判別.

・電池によるバックアップ.

(30) **拡張用スロット**

・FDC，HDC を含む拡張ボードを実装するためのスロット.

・PC-98XA の場合，HDC および 640KB FD インターフェイスボード(PC-9801-09)は，それぞれ専用のスロットに挿入.

・XA に ROM を搭載した 9800 シリーズ用のオプションボードを実装する場合，ROM Kill の設定が必要.

(31) **オプション**

・拡張用スロットに挿入されるオプション

a) 増設 RAM ボード(PC-98XL-01)

・512K バイトのメモリが実装されたボード，PC-98XL でのみ使用. 最初の増設時には必須.

b) 増設 RAM ボード(PC-9801-51)

・1M バイトのメモリが実装されたボード.

・PC-98XL の場合，PC-98XL-01 増設後，本ボードにより増設可能.

c) 増設 RAM ボード(PC-98XA-01K)

・256K バイトのメモリが実装されたボード.

・増設 RAM サブボード(PC-98XA-02K)を追加することにより，256K バイト単位で 1M バイトまで増設可能.

・PC-98XL(ハイレゾリューションモード)では，クロック 8MHz でのみ使用可能.

d) ユニバーサルボード(PC-9801-04)

・オリジナル回路作成用ボード.

e) 640KB FD インターフェイスボード(PC-9801-09)

・640KB FD インターフェイスボード.

・PC-98XL では，ノーマルモード時は，μPD70116 の 8MHz 動作時，ハイレゾリューションモード時は 8MHz 動作時のみで使用可能.

・PC-98XA では専用スロットに挿入.

f) GP-IBインターフェイスボード(PC-9801-29K)

・コンピュータ，計測器の標準的インターフェイスであるGP-IBインターフェイス用のボード．

g) RS-232C(第2－3回線用)拡張インターフェイスボード(RC-9861K)．

・RS-232Cインターフェイスを2回線拡張するためのボード．

h) 通信制御アダプタ(PC-9862/PC-9866)

・HDLCまたはSDLC手順の回線インターフェイスボード．

i) ネットワークインターフェイスセット(PC-9864)

・BRANCH-4670用インターフェイスボードとブランチケーブルのセット．

# 第3章

# CPU

## 3.1　CPU

・使用LSI

PC-98XA : i80286

PC-98XL : i80286, μPD70116

・CPUアクセス時のウエイト数

<table>
<tr><td colspan="2" rowspan="2">機種</td><td colspan="5">PC-98XL</td><td>PC-98XA</td></tr>
<tr><td colspan="3">ノーマル</td><td colspan="2">ハイレゾリューション</td><td></td></tr>
<tr><td></td><td>動作CPU</td><td colspan="2">70116</td><td>80286</td><td colspan="2">80286</td><td>80286</td></tr>
<tr><td></td><td>クロック</td><td>8MHz</td><td>10MHz</td><td>8MHz</td><td>8MHz</td><td>10MHz</td><td>8MHz</td></tr>
<tr><td rowspan="2">RAM</td><td>標準内蔵</td><td>1(5)</td><td>1(5)</td><td>0(2)</td><td>1(3)</td><td>1(3)</td><td rowspan="4">1(3)</td></tr>
<tr><td>1MB以上</td><td>アクセス不可</td><td>アクセス不可</td><td>1(3)</td><td>1(3)</td><td>1(3)</td></tr>
<tr><td rowspan="2">ROM</td><td>標準内蔵</td><td>1(5)</td><td>1(5)</td><td>0(2)</td><td>1(3)</td><td>1(3)</td></tr>
<tr><td>バンク0C, 0D</td><td>2(6)</td><td>3(7)</td><td>4(6)</td><td>—</td><td>—</td></tr>
<tr><td colspan="2">I/O</td><td>2(6)</td><td>3(7)</td><td>3(5)</td><td>3(5)</td><td>4(6)</td><td>3(5)</td></tr>
<tr><td colspan="2">INTA</td><td>1(5)</td><td>2(6)</td><td>5(7)</td><td>5(7)</td><td>5(7)</td><td>2(4)</td></tr>
</table>

(　)内は全クロック数

・CPUおよび数値データプロセッサの初期化は，システムリセット時や，復旧できないエラーが発生したときに行われるが，その他ソフトウエア(OUT命令によるシャットダウン)によっても実行できる．

・80286のプロテクトモードからリアルモードに戻る場合には，シャットダウンを利用する．

・CPU に対する命令

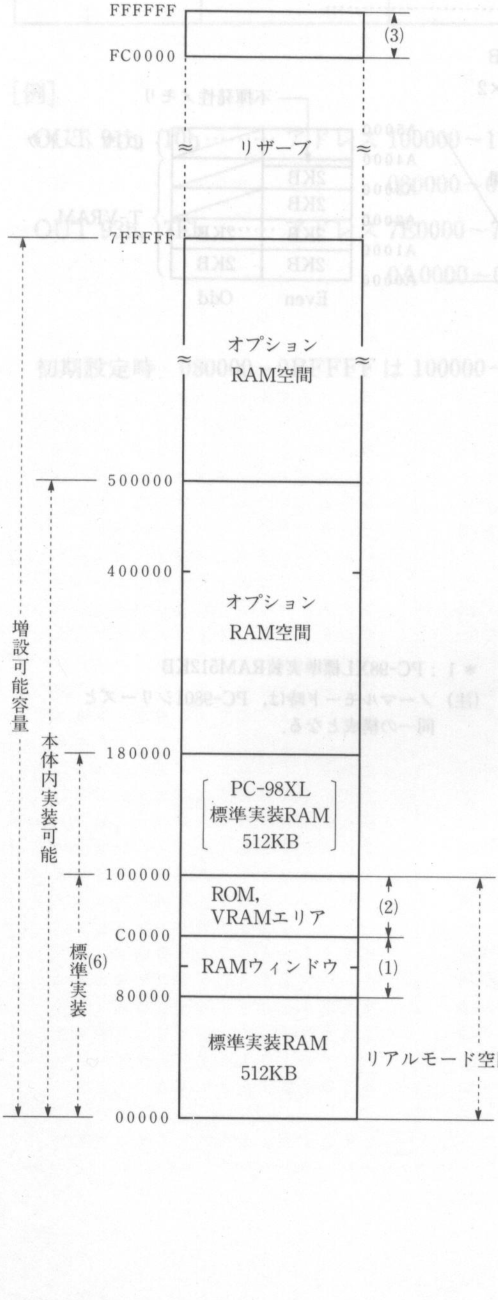
| 命　令 | ポート番号 | R/W | データ $b_7$ $b_6$ $b_5$ $b_4$ $b_3$ $b_2$ $b_1$ $b_0$ | 機　能 |
|---|---|---|---|---|
| Shut Doun | 00F0H | W | 0 0 0 0 0 0 0 0 | CPU，NDP を初期化する．<br>復旧できないエラーの対応，80286CPU 動作時，プロテクトモードからリアルモードへの移行に使用． |
| AB20 Enable | 00F2H | W | 0 0 0 0 0 0 0 0 | アドレスバス20ビット目のマスクを解除する（80286プロテクトモード）． |

＊上記命令は，80286CPU が動作しているときのみ有効．70116CPU が動作している時は命令を実行させてはならない．実行させた時の動作は保証されない．

# 第4章

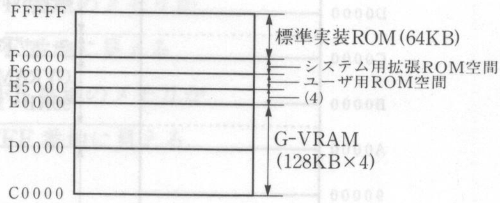
# メモリマップ

## 4.1 PC-98XA/XLハイレゾリューションモード時



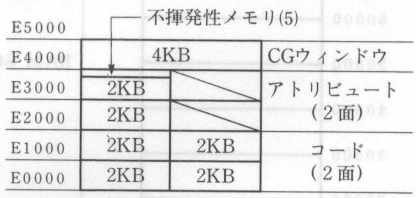

(1) RAMウィンドウ
80286CPUリアルモードにおいて，1MB以上の空間を参照するための窓である．従ってこの空間のみのアドレスを持つメモリは存在しない．

(2) ROM，VRAMエリア

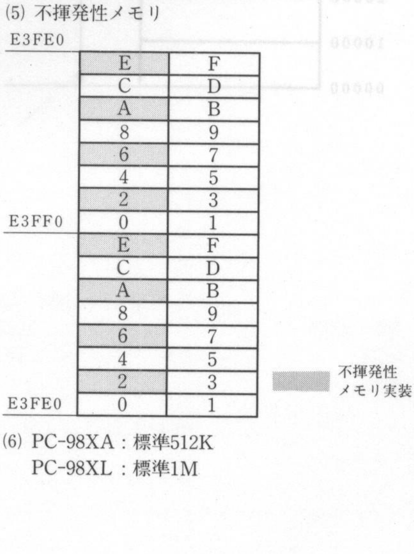



(3) FC0000~FFFFFF
0C0000~0FFFFFと同一の内容である．
(どちらのアドレス指定にも応じる．)

(4) 0E0000~0E4FFF

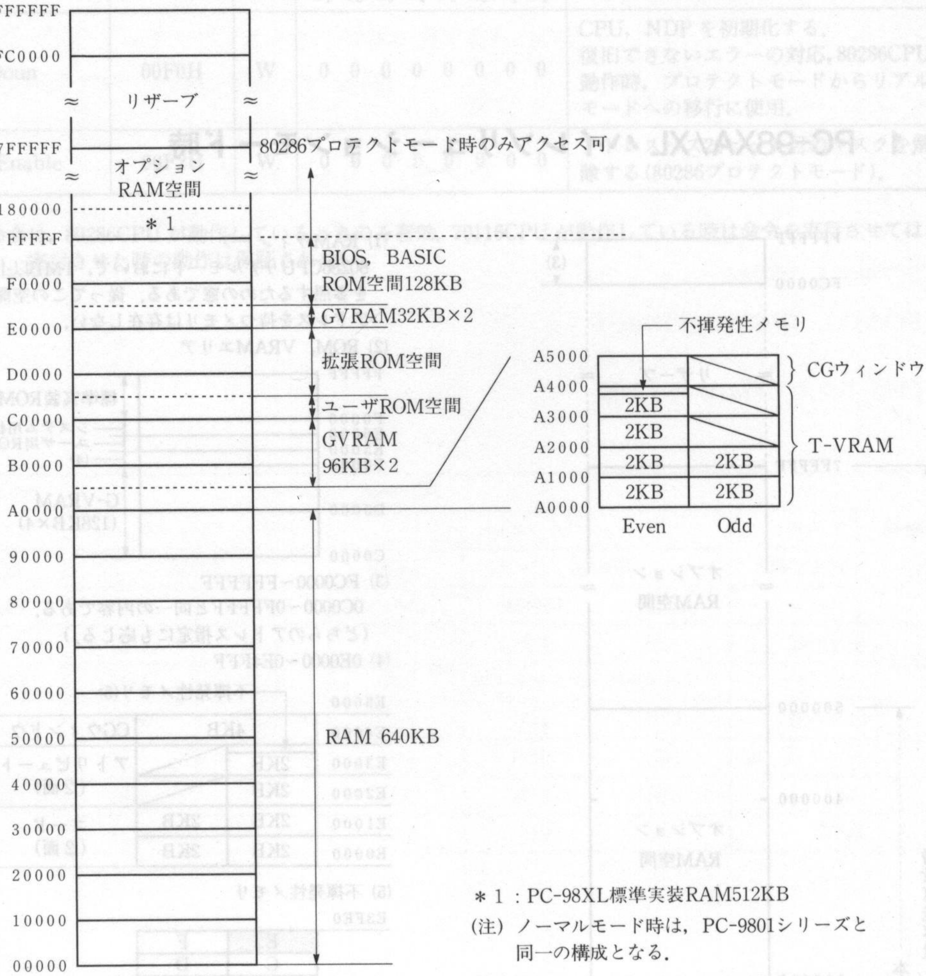



(5) 不揮発性メモリ




(6) PC-98XA：標準512K
PC-98XL：標準1M

## 4.2　PC-98XL ノーマルモード時




＊1：PC-98XL標準実装RAM512KB

(注) ノーマルモード時は，PC-9801シリーズと同一の構成となる．

## 4.3 RAMウインドウ

PC-98XAおよびPC-98XLのハイレゾリューションモードで，80286CPUのリアルモードにおいて1MB以上の空間を参照する場合は，RAMウインドウを使用する．

| コマンド | ポートアドレス | $b_7$ $b_6$ $b_5$ $b_4$ $b_3$ $b_2$ $b_1$ $b_0$ | 機能 |
|---|---|---|---|
| Set Bank 08 | 091H | A23 A22 A21 A20 A19 A18 A17 × | バンク08,09に見せる<br>RAMのバンクアドレス |
| Set Bank 0A | 093H | 〃 | バンク0A，0Bに見せる<br>RAMのバンクアドレス |

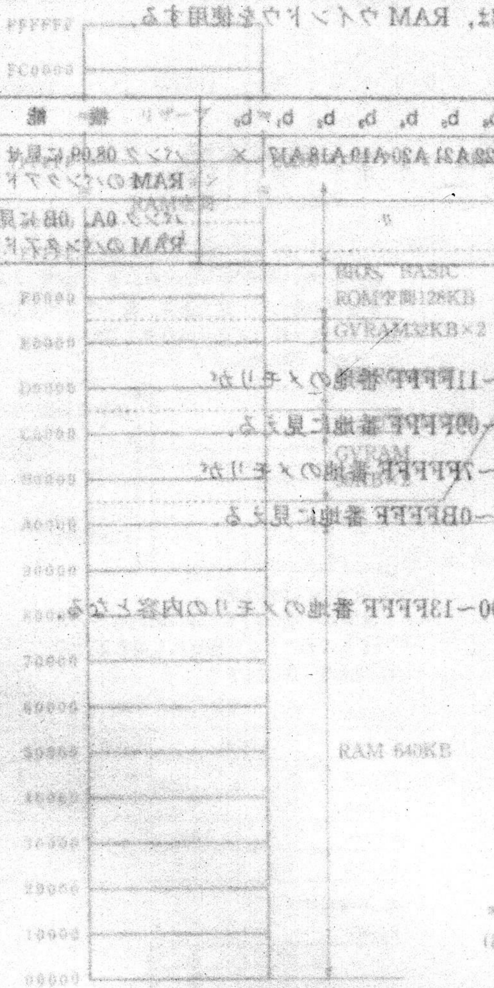
[例]

OUT 91h，10h……… アドレス100000～11FFFF番地のメモリが
080000～09FFFF番地に見える．

OUT 93h，7Eh …… アドレス7E0000～7FFFFF番地のメモリが
0A0000～0BFFFF番地に見える．

初期設定時　080000～0BFFFFは100000～13FFFF番地のメモリの内容となる．

## 4.2 PC-98XLノーマルモード時




＊1：PC-98XL標準実装RAM512KB
(注) ノーマルモード時は、PC-9801シリーズと同一の構成となる。

# 第5章

# 入出力制御

## 5.1 I/Oポートアドレス

| ポートアドレス<br>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | 装置名 | | |
|---|---|---|---|
| ××××00××00000×A0 | 割り込みコントローラ(M) | | 8259A |
| ××××00××00001×A0 | 割り込みコントローラ(S) | | 8259A |
| ××××00××000AAAA1 | DMAコントローラ | | 8237A |
| ××××00××0010×××0 | カレンダ時計 | | 4990 |
| ××××00××0010×AA1 | DMAバンク | | |
| ××××00××0011××A0 | RS-232Cインターフェイス | | 8251A |
| ××××00××0011×AA1 | システムポート | | 8255A |
| ××××01××00111AA1 | 予　約 | (注2) | |
| ××××00××0100×AA0 | プリンタインターフェイス | | 8255A |
| ××××00××0100××A1 | キーボードインターフェイス | | 8251A |
| ××××00××0101××A0 | NMIコントロール | | |
| ××××00××0101×AA1 | 予　約 | (注1) | |
| ××××00××0110AAA0 | CRTコントローラ(テキスト) | | 7220A |
| ××××00××0110×××1 | マウスインターフェイス | (注1)(注4) | 8255A |
| ××××00××0111AAA0 | CRTコントローラ | (注1) | 52611 |
| ××××00××0111×AA1 | タイマコントローラ | | 8253C |
| ××××00××10000AA0 | 固定ディスクインターフェイス | (注2) | |
| ××××××××100000A0 | 固定ディスクインターフェイス | (注1) | |
| ××××××××100001×0 | 予　約 | (注1) | |
| ××××00××10000××1 | 予　約 | (注2) | |
| ××00000A10001AA0 | MUSICインターフェイス | | |
| ××××××1100001AA0 | サウンドインターフェイス | | YM2203 |
| ××××××××10001AA1 | BRANCH4670 | | |
| ××××××××1001×AA0 | 1MB FDDインターフェイス | | 765A |
| ××××00××100100A1 | メモリウィンドウ | (注1) | |
| ××××0000100100A1 | メモリウィンドウ | (注2) | |
| ××××××××100110A1 | GP-IBスイッチ | | |
| ××××××××10011×1 | 予　約 | | |
| ××××00××1010AAA0 | CRTコントローラ(グラフ) | | 7220 |
| ××××01××1010AAA0 | EGC拡張アドレス | (注2) | |
| ××××00××1010AAA1 | 文字パターンROM | | |
| ××××××××10110AA0 | HDLC/SDLC | | 7201 |
| ××××××××10111××0 | 予　約 | | |
| ××××000010111110 | 1MB/640KB FDD切り換え | | |
| ××××××××1011×××1 | HDLC/SDLC | | 8253/8255 |
| ××××00××11001AA0 | 640KB FDD | | 765A |
| ××××××××1100AAA1 | GP-IB | | 7210 |
| ××××××××1101×××× | 未使用 | | |
| ××××××××11100000<br>∫<br>××××××××11101011 | キーボード(スキャン方式)予約 | | |

| ポートアドレス<br>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | 装　置　名 | | |
|---|---|---|---|
| ××××××××111011×× | 未使用 | | |
| ××××××××11110AA0 | CPU 制御 | | |
| ××××××××11110××1 | 未使用 | | |
| ××××00××11111AA0 | NDP | | |
| 0111111111011AA1 | マウス | (注3) | 8255 |
| 0011111111011011 | 内部サウンド周波数設定 | (注3) | 8253 |
| 1011111111011011 | マウス割り込み TIME 設定 | | |
| 1011111111011000 | 予　約 | | |
| 1011111111011010 | 予　約 | | |

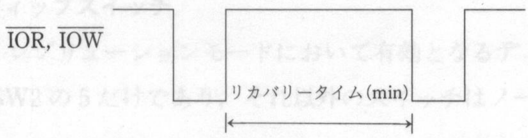
注1：PC-98XA
注2：PC-98XL
注3：PC-98XL，ノーマルモード
注4：PC-98XL，ハイレゾリューションモード

PC-98XL では，CMT インターフェイス I/O ポートとメモリウィンドウポートが同一のため，CMT はサポートしない．

## 5.2 周辺 LSI の連続アクセスについての制限事項

CPU から周辺 LSI に対して連続したアクセス(OUT, IN 命令の実行)を行うとき，最初のアクセスに対して周辺 LSI の動作が完了しない状態で,CPU からの次のアクセスが行われる場合がある．このようなことが起こると正常な動作とならず，結果として誤動作になる．正常な動作を保証するためには，最初の I/O 命令実行直後に，CPU 側で必要なだけのクロック数を JMP 命令でとり，周辺 LSI の動作完了を待って次の I/O 命令を実行する必要がある．PC-98XL/XA の80286動作において，周辺 LSI の動作完了までに必要な JMP 命令の数は次図のようになる．

なお PC-98XL における 70116CPU 動作時は，I/O 命令間に NOP を挿入して調整するが，動作完了までに要する NOP 命令の数は PC-9801VM と同様となる．数値については「PC-9800 シリーズテクニカルデータブック」を参照のこと．また，PC-98XL ノーマルモード時に，70116 と80286の 2 つの CPU を使用する場合があるが，Wait の差異は，ハードウェアによって吸収される．




単位 JMP$＋2 挿入数

| 周辺LSI名 | | リカバリータイム (nSEC) | WR → WR 8MHz | WR → WR 10MHz | RD → RD 8MHz | RD → RD 10MHz | WR → RD 8MHz | WR → RD 10MHz | RD → WR 8MHz | RD → WR 10MHz |
|---|---|---|---|---|---|---|---|---|---|---|
| 8237 | DMAC | 400 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 8253 | タイマ | 1000 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 |
| 8255A | PIO | 850 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 |
| 8259A | PIC | 190 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 8251A SIO | モード初期化 | （注3） | 3 | 3 | | | | | | |
| | ライトデータ非同期 | （注3） | 4 | 4 | | | | | | |
| | ライトデータ同期 | （注3） | 6 | 7 | | | | | | |
| | その他 | （注5） | | | 0 | 0 | 0 | 0 | 0 | 0 |
| 765AC | FDC | （注4）（注5） | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7220A | グラフ | 標準CRT 2020 | 2 | 3 | 2 | 2 | 2 | 3 | 2 | 2 |
| | 2.5Mモード | 高解像CRT 1210 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 1 |
| | テキスト/グラフ | 標準CRT 760 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 5Mモード | 高解像CRT 360 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| | グラフ2.5M（注1） | 2240 | 3 | 3 | 2 | 2 | 3 | 3 | 2 | 2 |
| | テキスト（注1） | 1120 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 1 |
| | グラフ（注2） | 670 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | テキスト（注2） | 600 | 1 | 1 | 0（1） | 1 | 1 | 1 | 0（1） | 1 |
| 7201 | 通信制御 | 300 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 7210 | GPIB | 250 | 1（0） | 1 | 0 | 0 | 1（0） | 1 | 0 | 0 |

カッコ(　)内の数値は PC-98XA
注1：スーパーインポーズ時
注2：ハイレゾリューションモード時
注3：CPU クロックと8251用クロックは，8/10MHz モード切り替えにより同時に変化する.
注4：765A ステータスの ROM/DIO の指示により I/O 動作を行う限り特に制約はない.
注5：LSI として制限がないので連続アクセス可である.

# 5.3 スイッチ

## 5.3.1 PC-98XL

### (1) クロック切り換えスイッチ

| 設　定 | クロック |
|---|---|
| 8 | 8MHz |
| 10 | 10MHz |

既定値

ノーマルモードで80286CPU動作時は，クロック切り替えスイッチは無視され，常に8MHzとなる.

### (2) モード切り替えスイッチ

| 設定 | モード名 | 使用CPU |
|---|---|---|
| N | ノーマルモード | 70116または80286 |
| H | ハイレゾリューションモード | 80286 |

既定値

ノーマルモード時の使用CPUは，ディップスイッチSW3-8の設定により選択される.

### (3) ディップスイッチ

ハイレゾリューションモードにおいて有効となるディップスイッチは，SW1の4，5，6およびSW2の5だけであり，それ以外のスイッチはノーマルモードでのみ意味を持つ．

| スイッチ名 | 番号 | 意味 ON | 意味 OFF | ソフトウェアからの参照ポート |
|---|---|---|---|---|
| SW1 | 1 | 専用高解像度ディスプレイ | 標準ディスプレイ | システム　B3 |
| | 2 | スーパーインポーズ使用 | スーパーインポーズ未使用 | |
| | 3 | プラズマディスプレイ使用 | プラズマディスプレイ未使用 | プリンタ　B4 |
| | 4 | 内蔵FD番号　#3，#4 | 内蔵FD番号　#1，#2 | FD |
| | 5 | RS-232C　同期モード | 選択（注1） | マウス　C1 |
| | 6 | RS-232C　同期モード | 選択（注1） | 〃　C2 |
| | 7 | | 常にOFF | |
| | 8 | 拡張グラフィックモード | 基本グラフィックモード | プリンタ　B3 |
| SW2 | 1 | 不可 | 常にOFF | システム　A0 |
| | 2 | ターミナルモード | BASICモード | 〃　A1 |
| | 3 | 80字/行 | 40字/行 | 〃　A2 |
| | 4 | 25行 | 20行 | 〃　A3 |
| | 5 | メモリスイッチ保持 | メモリスイッチ初期化 | 〃　A4 |
| | 6 | 内蔵固定ディスク切り離し＊ | | 〃　A5 |
| | 7 | | 常にOFF | 〃　A6 |
| | 8 | GDC　5 MHzモード | GDC　2.5MHzモード | 〃　A7 |
| SW3 | 1 | 内蔵FD固定モード | プログラムモード | |
| | 2 | 内蔵FD　640KBモード | 内蔵FD　1 MBモード | |
| | 3 | | 常にOFF | |
| | 4 | | 〃 | |
| | 5 | | 〃 | |
| | 6 | | 〃 | |
| | 7 | | 〃 | |
| | 8 | 動作CPU　80286 | 動作CPU　70116 | プリンタ　B1 |

既定値　　＊mode14のみ

注1：同期モード選択

| スイッチ番号 5 | スイッチ番号 6 | 同期モード |
|---|---|---|
| ON | ON | BCI同期 |
| ON | OFF | ST2同期 |
| OFF | ON | 同期刻時機構 |
| OFF | OFF | 調歩同期 |

既定値

### (4) メモリスイッチ

PC-98XL では，2組のメモリスイッチを持っている．スイッチの内容は，ハイレゾリューションモード時は PC-98XA と同様であり，ノーマルモード時は PC-9801VM2と同様である．ただし，ノーマルモード時の標準実装メモリ容量は 640KB が既定値であり，数値データプロセッサ関係の設定が次の様になっている．

SW3-3：80286使用時の数値演算プロセッサの有無．
SW3-4：70116使用時の数値演算プロセッサの有無．
SW3-5：数値演算プロセッサのクロック周波数．
　0-10MHz
　1- 8MHz

なお，SW3-5 は，MS-DOS においてのみ機能する．

## 5.3.2 PC-98XA

### (1) ディップスイッチ

| スイッチ名 | 番号 | 意味 ON | 意味 OFF | ソフトウェアからの参照ポート | |
|---|---|---|---|---|---|
| SW1 | 1 | | 常に OFF | システム | A0 |
| | 2 | | 常に OFF | 〃 | A1 |
| | 3 | | 常に OFF | 〃 | A2 |
| | 4 | | 常に OFF | 〃 | A3 |
| | 5 | メモリスイッチ保持 | メモリスイッチ初期化 | 〃 | A4 |
| | 6 | | 常に OFF | 〃 | A5 |
| | 7 | | 常に OFF | 〃 | A6 |
| | 8 | | 常に OFF | 〃 | A7 |
| | 9 | 選択（注1） | RS-232C　同期モード | マウス | C0 |
| | 10 | 選択（注1） | RS-232C　同期モード | 〃 | C1 |
| SW2 | 1 | 常に ON | | マウス | B0 |
| | 2 | 選択（注2） | 拡張メモリサイズ | 〃 | B1 |
| | 3 | 選択（注2） | 拡張メモリサイズ | 〃 | B2 |
| | 4 | 選択（注2） | 拡張メモリサイズ | 〃 | B3 |
| | 5 | 選択（注2） | 拡張メモリサイズ | 〃 | B4 |
| | 6 | 選択（注2） | 拡張メモリサイズ | 〃 | B5 |
| | 7 | | 常に OFF | 〃 | B6 |
| | 8 | 内蔵 FD 番号　#3，#4 | 内蔵 FD 番号　#1，#2 | 〃 | B7 |

▇ 既定値

注1：RS-232C　同期モード選択

| スイッチ番号 | | 同期モード |
|---|---|---|
| 9 | 10 | |
| OFF | OFF | BCI 同期 |
| OFF | ON | ST2 同期 |
| ON | OFF | 同期刻時機構 |
| ON | ON | 調歩同期 |

■ 既定値

注2：拡張メモリサイズ
標準実装（512KB）時を0 KBとし，メモリ拡張時に以下の設定を行う．

| SW2…SW6 | 拡張メモリサイズ | SW2…SW6 | 拡張メモリサイズ |
|---|---|---|---|
| 0 0 0 0 0 | 0KB | 0 1 1 1 1 | 3840KB |
| 0 0 0 0 1 | 256KB | 1 0 0 0 0 | 4096KB |
| 0 0 0 1 0 | 512KB | 1 0 0 0 1 | 4352KB |
| 0 0 0 1 1 | 768KB | 1 0 0 1 0 | 4608KB |
| 0 0 1 0 0 | 1024KB | 1 0 0 1 1 | 4864KB |
| 0 0 1 0 1 | 1280KB | 1 0 1 0 0 | 5120KB |
| 0 0 1 1 0 | 1536KB | 1 0 1 0 1 | 5376KB |
| 0 0 1 1 1 | 1792KB | 1 0 1 1 0 | 5632KB |
| 0 1 0 0 0 | 2048KB | 1 0 1 1 1 | 5888KB |
| 0 1 0 0 1 | 2304KB | 1 1 0 0 0 | 6144KB |
| 0 1 0 1 0 | 2560KB | 1 1 0 0 1 | 6400KB |
| 0 1 0 1 1 | 2816KB | 1 1 0 1 0 | 6656KB |
| 0 1 1 0 0 | 3072KB | 1 1 0 1 1 | 6912KB |
| 0 1 1 0 1 | 3328KB | 1 1 1 0 0 | 7168KB |
| 0 1 1 1 0 | 3584KB | | |

0：ON　1：OFF

## (2) メモリスイッチ

PC-98XAのMS-DOS環境においては，以下のメモリスイッチを使用している．

| スイッチ名（メモリ番地） | データ（ビット位置） 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 機能 | |
|---|---|---|---|---|---|---|---|---|---|---|
| SW1 (E3FE2) | | | | | | | | 0 | Xパラメータ | 有効 |
| | | | | | | | | 1 | | 無効 |
| | | | | | 1 | 0 | | | データビット長 | 7ビット長 |
| | | | | | 1 | 1 | | | | 8ビット長 |
| | | | | 0 | | | | | パリティチェック | あり |
| | | | | 1 | | | | | | なし |
| | | | 0 | | | | | | パリティ指定 | 奇数 |
| | | | 1 | | | | | | | 偶数 |
| | 0 | 1 | | | | | | | ストップビット長 | 1ビット |
| 既定値 48H | 1 | 1 | | | | | | | | 2ビット |

| スイッチ名<br>（メモリ番地） | データ（ビット位置） | | | | | | | | 機能 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| | | | | | 0 | 0 | 0 | 0 | 通信速度 | 無効 |
| | | | | | 0 | 0 | 0 | 1 | | 75bps |
| | | | | | 0 | 0 | 1 | 0 | | 150bps |
| | | | | | 0 | 0 | 1 | 1 | | 300bps |
| SW2<br>(E3FE4) | | | | | 0 | 1 | 0 | 0 | | 600bps |
| | | | | | 0 | 1 | 0 | 1 | | 1200bps |
| | | | | | 0 | 1 | 1 | 0 | | 2400bps |
| | | | | | 0 | 1 | 1 | 1 | | 4800bps |
| 既定値 05H | | | | | 1 | 0 | 0 | 0 | | 9600bps |

| スイッチ名<br>（メモリ番地） | データ（ビット位置） | | | | | | | | 機能 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| | | | | | | 0 | 0 | 0 | メモリサイズ | 128KB |
| | | | | | | 0 | 0 | 1 | | 256KB |
| | | | | | | 0 | 1 | 0 | | 384KB |
| | | | | | | 0 | 1 | 1 | | 512KB |
| SW3<br>(E3FE6) | | | | | | 1 | 0 | 0 | | 640KB |
| | | | | | | 1 | 0 | 1 | | 768KB |
| | | | | | 0 | | | | NDPの有無 | 無し |
| | | | | | 1 | | | | | 有り |
| | | 0 | | | | | | | テキスト画面の初期色指定 | 白 |
| 既定値 03H | | 1 | | | | | | | | 緑 |

| スイッチ名<br>（メモリ番地） | データ（ビット位置） | | | | | | | | 機能 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| SW4<br>(E3FE8) | | | | 0 | | | | | ネットワークインタフェイスの有無 | 無し |
| 既定値 00H | | | | 1 | | | | | | 有り |

| スイッチ名<br>（メモリ番地） | データ（ビット位置） | | | | | | | | 機能 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| SW5<br>(E3FEA) | | | | | | | | 0 | プリンタ種別 | 16ドット系 |
| 既定値 00H | | | | | | | | 1 | | 24ドット系 |

# 第6章

# 割り込みベクタ

PC-98XA および PC-98XL のハイレゾリューションモード時は，下図のようにベクタを使用する．PC-98XL のノーマルモード時は PC-9800 シリーズと同様となる．「PC-9800 シリーズテクニカルデータブック」参照のこと．

| ベクタアドレス | ベクタ番号 | 用途 | 備考 |
|---|---|---|---|
| 0 - 3 | 0 | 0 除算エラー | |
| 4 - 7 | 1 | シングルステップ | |
| 8 - B | 2 | NMI | |
| C - F | 3 | INT3 | |
| 10 - 13 | 4 | オーバーフロー | |
| 14 - 17 | 5 | コピー(COPY)キー | キーは変更可能 |
| 18 - 1B | 6 | STOP キー | キーは変更可能 |
| 1C - 1F | 7 | インターバルタイマ | |
| 20 - 23 | 8 | タイマ | |
| 24 - 27 | 9 | キーボード | |
| 28 - 2B | A | CRTV(V-SYNC) | |
| 2C - 2F | B | 拡張バス　INT0 | |
| 30 - 33 | C | RS-232C（ch0） | |
| 34 - 37 | D | 拡張バス　INT1 | |
| 38 - 3B | E | マウス（拡張バス　INT2） | |
| 3C - 3F | F | システム予約 | |
| 40 - 43 | 10 | NDP(80287) | |
| 44 - 47 | 11 | HD，HDLC | （注 1） |
| 48 - 4B | 12 | 640KBFD | （注 1） |
| 4C - 4F | 13 | 1MBFD | （注 1） |
| 50 - 53 | 14 | RS-232C（ch1，ch2） | |
| 54 - 57 | 15 | MUSIC | |
| 58 - 5B | 16 | プリンタ | |
| 5C - 5F | 17 | ノイズ | |
| 60 - 63 | 18 | KB/CRT BIOS | |
| 64 - 67 | 19 | RS-232C BIOS | |
| 68 - 6B | 1A | プリンタ BIOS | |
| 6C - 6F | 1B | DISK BIOS | |
| 70 - 73 | 1C | カレンダ BIOS | |
| 74 - 77 | 1D | GRAPH BIOS | |
| 78 - 7B | 1E | | |
| 7C - 7F | 1F | システム（BIOS） | |
| 80～FF | 20～3F | システム予約 | |
| 100～1FF | 40～7F | 未使用 | Boot 処理の間，システムスタックとして使用される |

注 1：装置が接続されている時のみベクタがセットされる．
注 2：ベクタ番号 8～17 はハードウェア割り込みに使用．

# 第6章 割り込みベクタ

PC-98XA および PC-98XL²のハイレゾリューションモード時は、下図のようにベクタを使用する。PC-98XL²のノーマルモード時は PC-9800 シリーズと同様となる。「PC-9800 シリーズテクニカルデータブック」参照のこと。

| ベクタアドレス | ベクタ番号 | 要因 | 備考 |
|---|---|---|---|
| 0 - 3 | 0 | 0除算エラー | |
| 4 - 7 | 1 | シングルステップ | |
| 8 - B | 2 | NMI | |
| C - F | 3 | INT3 | |
| 10 - 13 | 4 | オーバーフロー | |
| 14 - 17 | 5 | ハードコピー(COPY)キー | キーは変更可能 |
| 18 - 1B | 6 | STOPキー | キーは変更可能 |
| 1C - 1F | 7 | インターバルタイマ | |
| 20 - 23 | 8 | タイマ | |
| 24 - 27 | 9 | キーボード | |
| 28 - 2B | A | CRT(V-SYNC) | |
| 2C - 2F | B | 拡張バス INT0 | |
| 30 - 33 | C | RS-232C (ch0) | |
| 34 - 37 | D | 拡張バス INT1 | |
| 38 - 3B | E | マウス(拡張バス INT2) | |
| 3C - 3F | F | システム予約 | |
| 40 - 43 | 10 | NDP(80287) | |
| 44 - 47 | 11 | HD, HDLC | |
| 48 - 4B | 12 | 640KBFD | |
| 4C - 4F | 13 | 1MBFD | |
| 50 - 53 | 14 | RS-232C (ch1, ch2) | |
| 54 - 57 | 15 | MUSIC | |
| 58 - 5B | 16 | プリンタ | |
| 5C - 5F | 17 | ノイズ | |
| 60 - 63 | 18 | KB/CRT BIOS | |
| 64 - 67 | 19 | RS-232C BIOS | |
| 68 - 6B | 1A | プリンタ BIOS | |
| 6C - 6F | 1B | DISK BIOS | |
| 70 - 73 | 1C | カレンダ BIOS | |
| 74 - 77 | 1D | GRAPH BIOS | |
| 78 - 7B | 1E | | |
| 7C - 7F | 1F | システム(BIOS) | |
| 80 - FF | 20 - 3F | システム予約 | |
| 100 - 1FF | 40 - 7F | 未使用 | Boot 処理の間、システムスタックとして使用される |

注1：装置が接続されている時のみベクタがセットされる。
注2：ベクタ番号8～17はハードウェア割り込みに使用。

# 第7章

# BOOT STRAP

## 7.1 媒体とIPLレコード形式，LOADアドレス

| システム立上げ装置 | システム立上げユニット番号 | IPLレコード媒体フォーマット(0シリンダ0トラック) 記録密度 | セクタ長 | IPLレコードセクタ番号 | IPLレコードバイト数 | IPLレコードのロードアドレス(実行エントリポイント) セグメントベース(×16) | オフセットアドレス | 備考 |
|---|---|---|---|---|---|---|---|---|
| 640KB FD | 0,1,2,3 | FM | 128 | 1，2，3，4 | 512 | 1FE0H | 0000H | |
| | | | 256 | 1，2 | | | | |
| | | | 512 | 1 | | | | |
| | | MFM | 256 | 1，2，3，4 | 1024 | 1FC0H | | |
| | | | 512 | 1，2 | | | | |
| | | | 1024 | 1 | | | | |
| 1MB FD | 0,1,2,3 | FM | 128 | 1，2，3，4 | 512 | 1FE0H | | |
| | | | 256 | 1，2 | | | | |
| | | | 512 | 1 | | | | |
| | | MFM | 256 | 1，2，3，4 | 1024 | 1FC0H | | |
| | | | 512 | 1，2 | | | | |
| | | | 1024 | 1 | | | | |
| HD | 0，1 | MFM | 256 | 0，1 | 512 | 1FE0H | | IPLレコードトラック番号は論理ボリューム相対 |
| | | | 512 | | 1024 | | | |

注:PC-98XLノーマルモード時は上記とは異なる．

## 7.2　BOOT 処理

ハードウェア試験診断実行後，ブート処理を行う．

メモリ SW の状態に従って，ブート可能な装置の IPL を探し，見つかった IPL に制御を渡す．Power ON または Reset 時にファンクションキーが押下されていた場合は，メモリスイッチをファンクションキーの値によって置き換え，固定ディスクの IPL を探し，見つかった IPL に制御を渡す．

以上の処理を行っても IPL が見つからない場合は，すべてのアクセス可能な DISK のサーチを行いながらアイドル状態となる．

**・BOOT ROM と IPL のインターフェイス**

① IPL エントリ時のレジスタ

AL　　＝DA/UA　(DISK BIO 参照)

DS　　＝0000H

SS:SP＝SYSTEM STACK

その他のレジスタは不定

② IPL から BOOT ROM へのリターン

SS:SP，DS の値を保証すれば，RETF で BOOT ROM へ戻ることができる．

BOOT ROM は IPL のサーチを続行する．

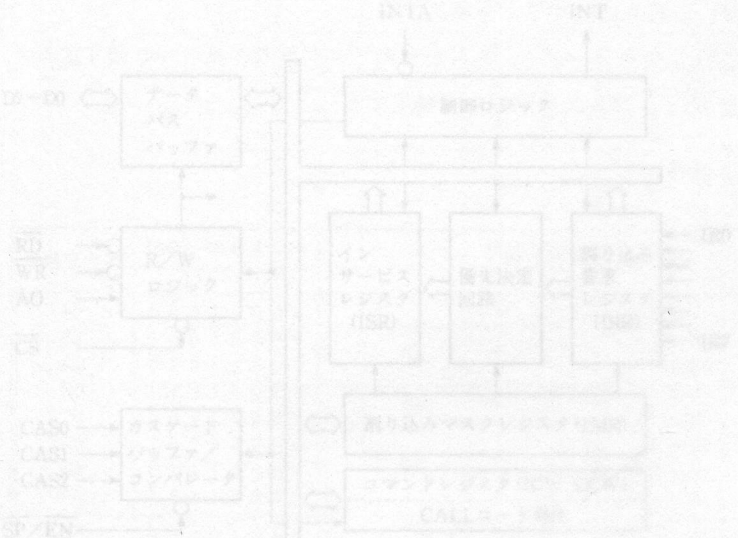
# 第1章

# 割り込みコントローラ

## 1.1 割り込みコントローラ(PIC)μPD8259A

μPD8259Aは割り込みルーチンアドレスを発生するため，割り込み要求デバイスをポーリングすることなく，要求のあった特定の割り込みルーチンへ直接/間接にジャンプすることができる．PC-98XL/XAではμPD8259Aを2個使用し，ハードウェア割り込みレベル15種(内1種システム使用)に対応している．

### (1) μPD8259Aの各ブロックの機能




端子名称

| | |
|---|---|
| $\overline{CS}$ | チップセレクト |
| $\overline{WR}$ | ライト入力 |
| $\overline{RD}$ | リード入力 |
| D7～D0 | データバス |
| CAS2～CAS0 | カスケード |
| $\overline{SP}/\overline{EN}$ | スレーブプログラム入力/イネーブル出力 |
| INT | 割り込み要求出力 |
| IR7～IR0 | 割り込み要求入力 |
| $\overline{INTA}$ | 割り込みアクノリッジ入力 |
| A0 | コマンド選択アドレス |

① IRRとISR

IR入力からの割り込みは，IRR(割り込み要求レジスタ)とISR(割り込みサービスレジスタ)の2つによって操作される．

IRRはサービスを要求しているすべての割り込みレベルを格納するのに用いられ，ISRはサービス中の割り込みレベルを格納するために用いられる．

② 優先決定回路

IRRにセットされたビットの優先順位を決定する．

最高位の優先ビットが選択され，$\overline{INTA}$パルス間にISRの該当ビットをセットする．

## 7.2 BOOT処理

ハードウェア試験診断実行後，ブート処理を行う。

メモリSWの状態に従って，ブート可能な装置のIPLを探し，見つかったIPLに制御を渡す．Power ONまたはReset時にファンクションキーが押下されていた場合は，メモリスイッチをファンクションキーの値によって置き換え，固定ディスクのIPLを探し，見つかったIPLに制御を渡す．

以上の処理を行ってもIPLが見つからない場合は，すべてのアクセス可能なDISKのサーチを行いながらアイドル状態となる．

・BOOT ROMとIPLのインターフェイス

①IPLエントリ時のレジスタ

AL　=DA/UA　(DISK BIO参照)

DS　=0000H

SS:SP=SYSTEM STACK

その他のレジスタは不定

②IPLからBOOT ROMへのリターン

SS:SP，DSの値を保存すれば，RETFでBOOT ROMへ戻ることができる．

BOOT ROMはIPLのサーチを続行する．

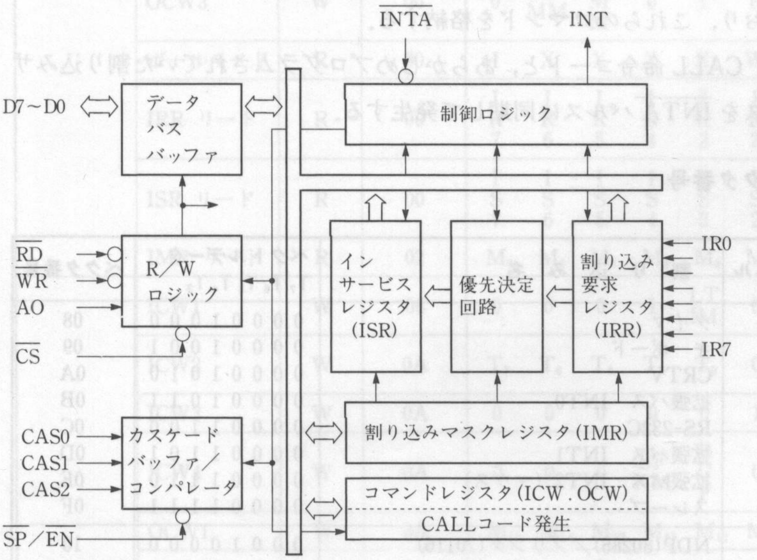
# 第1章

# 割り込みコントローラ

## 1.1 割り込みコントローラ(PIC)μPD8259A

μPD8259Aは割り込みルーチンアドレスを発生するため，割り込み要求デバイスをポーリングすることなく，要求のあった特定の割り込みルーチンへ直接/間接にジャンプすることができる．PC-98XL/XAではμPD8259Aを2個使用し，ハードウェア割り込みレベル15種(内1種システム使用)に対応している．

### (1) μPD8259Aの各ブロックの機能




| 端子名称 | |
|---|---|
| CS | :チップセレクト |
| WR | :ライト入力 |
| RD | :リード入力 |
| D7−D0 | :データバス |
| CAS2−CAS0 | :カスケード |
| SP／EN | :スレーブプログラム入力／イネーブル出力 |
| INT | :割り込み要求出力 |
| IR7−IR0 | :割り込み要求入力 |
| INTA | :割り込みアクノリッジ入力 |
| A0 | :コマンド選択アドレス |

① IRRとISR

IR入力からの割り込みは，IRR(割り込み要求レジスタ)とISR(割り込みサービスレジスタ)の2つによって操作される．

IRRはサービスを要求しているすべての割り込みレベルを格納するのに用いられ，ISRはサービス中の割り込みレベルを格納するために用いられる．

② 優先決定回路

IRRにセットされたビットの優先順位を決定する．

最高位の優先ビットが選択され，INTAパルス間にISRの該当ビットをセットする．

③ IMR

IMR(割り込みマスクレジスタ)はマスクすべき割り込みラインのビットを格納している。IMRのセットされているビットに対応したIR入力の割り込み要求は保留される。

④ データバスバッファ

コマンドワードとステータス情報をデータバスによって転送するための8ビットバッファである。

⑤ リード／ライトコントロールロジック

データバスを介してCPUから書き込まれるコマンドを受け取ったり，8259Aのステータスをデータバスに出力するための制御をする。

⑥ カスケードバッファ／コンパレータ

システム内で用いられる2つの8259AのIDを格納し，比較を行い，マスタ／スレーブとしての機能を制御する。

⑦ コマンドレジスタ，CALL発生ロジック

ICW(イニシャライズコマンドワード)レジスタ，OCW(オペレーションコマンドワード)レジスタの2つを含んでおり，これらのコマンドを格納する。

CALL発生ロジックは，CALL命令コードと，あらかじめプログラムされていた割り込みサービスルーチンアドレスを$\overline{INTA}$パルスに同期して発生する。

### (2) 割り込みレベルとベクタ番号

| デバイス名 | 割り込み要求信号 | レベル | 割り込み名 | ベクトルデータ $T_7 T_6 T_5 T_4 T_3$ | ベクタ番号 |
|---|---|---|---|---|---|
| | IR0 | 0 | タイマ | 0 0 0 0 1 0 0 0 | 08 |
| μPD8259A | IR1 | 1 | キーボード | 0 0 0 0 1 0 0 1 | 09 |
| | IR2 | 2 | CRTV | 0 0 0 0 1 0 1 0 | 0A |
| (マスタ) | IR3 | 3 | 拡張バス　INT0 | 0 0 0 0 1 0 1 1 | 0B |
| | IR4 | 4 | RS-232C | 0 0 0 0 1 1 0 0 | 0C |
| | IR5 | 5 | 拡張バス　INT1 | 0 0 0 0 1 1 0 1 | 0D |
| | IR6 | 6 | 拡張バス　INT2 (マウス) | 0 0 0 0 1 1 1 0 | 0E |
| | IR7 | 7 | スレーブ | 0 0 0 0 1 1 1 1 | 0F |
| | IR8 | 8 | NDP(80286)／プリンタ(70116) | 0 0 0 1 0 0 0 0 | 10 |
| μPD8259A | IR9 | 9 | 拡張バス　INT3 (固定ディスク) | 0 0 0 1 0 0 0 1 | 11 |
| | IR10 | 10 | 拡張バス　INT41 (640KB FD) | 0 0 0 1 0 0 1 0 | 12 |
| (スレーブ) | IR11 | 11 | 拡張バス　INT42 (1MB FD) | 0 0 0 1 0 0 1 1 | 13 |
| | IR12 | 12 | 拡張バス　INT5 | 0 0 0 1 0 1 0 0 | 14 |
| | IR13 | 13 | 拡張バス　INT6 | 0 0 0 1 0 1 0 1 | 15 |
| | IR14 | 14 | プリンタ(XA)/無(80286)/NDP(70116) | 0 0 0 1 0 1 1 0 | 16 |
| | IR15 | 15 | | 0 0 0 1 0 1 1 1 | 17 |

・マスタおよびスレーブの$T_7$～$T_3$は，それぞれ独立にプログラムで値をセットする。

・IR8およびIR14はCPUによって変化する。また，80287の割り込みは80286に直結されているため，80287の割り込みは8259では制御できない。

・レベル15の割り込みはなにもせずにリターンする。

・マウスの割り込みレベルは，PC-98XAおよびPC-98XLハイレゾリューションモード時ではINT2(IR6)に固定されているが，PC-98XLノーマルモード時はストラップスイッチによりINT0～INT6のいずれか選択可能(既定値はINT6)。

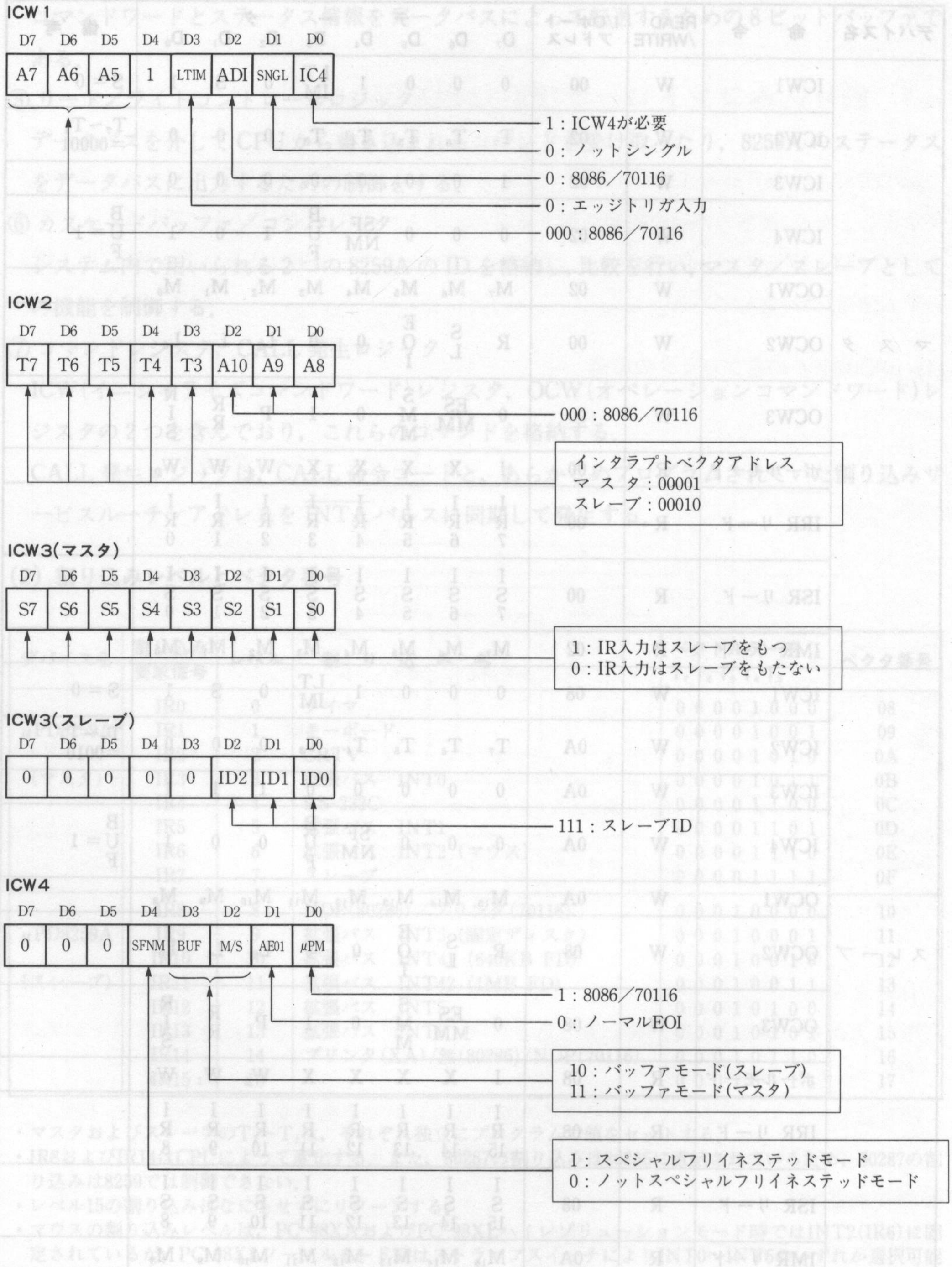
# 1.2 I/Oアドレスと命令

## (1) 割り込みコントローラ PIC$\mu$PD8259A のI/Oポートアドレスとデータ

| デバイス名 | 命令 | READ /WRITE | I/Oポート アドレス | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| マスタ | ICW1 | W | 00 | 0 | 0 | 0 | 1 | LTIM | 0 | S | 1 | S＝0 |
| | ICW2 | W | 02 | $T_7$ | $T_6$ | $T_5$ | $T_4$ | $T_3$ | 0 | 0 | 0 | $T_7$～$T_3$ ＝00001 |
| | ICW3 | W | 02 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | ICW4 | W | 02 | 0 | 0 | 0 | SFNM | BUF | 1 | 0 | 1 | BUF＝1 |
| | OCW1 | W | 02 | $M_7$ | $M_6$ | $M_5$ | $M_4$ | $M_3$ | $M_2$ | $M_1$ | $M_0$ | |
| | OCW2 | W | 00 | R | SL | EOI | 0 | 0 | $L_2$ | $L_1$ | $L_0$ | |
| | OCW3 | W | 00 | 0 | ESMM | SMM | 0 | 1 | P | RR | RIS | |
| | ポールモード | R | 00 | I | X | X | X | X | $W_2$ | $W_1$ | $W_0$ | |
| | IRR リード | R | 00 | IR7 | IR6 | IR5 | IR4 | IR3 | IR2 | IR1 | IR0 | |
| | ISR リード | R | 00 | IS7 | IS6 | IS5 | IS4 | IS3 | IS2 | IS1 | IS0 | |
| | IMR リード | R | 02 | $M_7$ | $M_6$ | $M_5$ | $M_4$ | $M_3$ | $M_2$ | $M_1$ | $M_0$ | |
| スレーブ | ICW1 | W | 08 | 0 | 0 | 0 | 1 | LTIM | 0 | S | 1 | S＝0 |
| | ICW2 | W | 0A | $T_7$ | $T_6$ | $T_5$ | $T_4$ | $T_3$ | 0 | 0 | 0 | $T_7$～$T_3$ ＝0010 |
| | ICW3 | W | 0A | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |
| | ICW4 | W | 0A | 0 | 0 | 0 | SFNM | BUF | 0 | 0 | 1 | BUF＝1 |
| | OCW1 | W | 0A | $M_{15}$ | $M_{14}$ | $M_{13}$ | $M_{12}$ | $M_{11}$ | $M_{10}$ | $M_9$ | $M_8$ | |
| | OCW2 | W | 08 | R | SL | EOI | 0 | 0 | $L_2$ | $L_1$ | $L_0$ | |
| | OCW3 | W | 08 | 0 | ESMM | SMM | 0 | 1 | P | RR | RIS | |
| | ポールモード | R | 08 | I | X | X | X | X | $W_2$ | $W_1$ | $W_0$ | |
| | IRR リード | R | 08 | IR15 | IR14 | IR13 | IR12 | IR11 | IR10 | IR9 | IR8 | |
| | ISR リード | R | 08 | IS15 | IS14 | IS13 | IS12 | IS11 | IS10 | IS9 | IS8 | |
| | IMR リード | R | 0A | $M_{15}$ | $M_{14}$ | $M_{13}$ | $M_{12}$ | $M_{11}$ | $M_{10}$ | $M_9$ | $M_8$ | |

### (2) イニシャライズコマンドワード(ICW)

ICWは$\mu$PD8259Aをイニシャライズするためのものである．$\mu$PD8259Aを機能させるためには，このコマンドワードによる設定を行う必要がある．



### (3) オペレーションコマンドワード(OCW)

イニシャライズの後は，いつでもOCWを書き込み，処理を求めることができる．OCWは，割り込みマスク，割り込み終了，優先回転，割り込みステータスを制御する．

**OCW1**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 |

インタラプトマスク　1：セット
　　　　　　　　　　0：リセット

**OCW2**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| R | SL | EOI | 0 | 0 | L2 | L1 | L0 |

| L2 | L1 | L0 | 作用されるIRレベル | |
|---|---|---|---|---|
| 0 | 0 | 0 | IR0 | IR8 |
| 0 | 0 | 1 | IR1 | IR9 |
| 0 | 1 | 0 | IR2 | IR10 |
| 0 | 1 | 1 | IR3 | IR11 |
| 1 | 0 | 0 | IR4 | IR12 |
| 1 | 0 | 1 | IR5 | IR13 |
| 1 | 1 | 0 | IR6 | IR14 |
| 1 | 1 | 1 | IR7 | IR15 |

| R | SL | EOI | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 非特殊EOI | 割り込み終了 |
| 0 | 1 | 1 | 特殊EOI* | |
| 1 | 0 | 1 | 非特殊EOIで回転 | 自動回転 |
| 1 | 0 | 0 | 自動EOIモードで回転(SET) | |
| 0 | 0 | 0 | 自動EOIモードで回転(CLEAR) | |
| 1 | 1 | 1 | 特殊EOIで回転* | 特殊回転 |
| 1 | 1 | 0 | 優先セットコマンド* | |
| 0 | 1 | 0 | ノーオペレーション | |

*：作用されるIRレベルを指定

**OCW3**

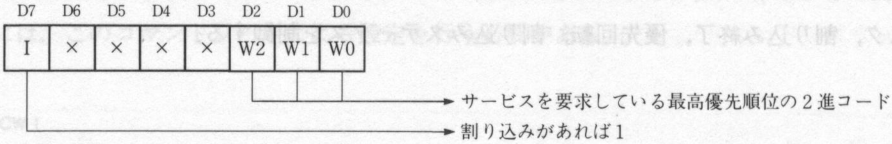
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| 0 | ESMM | SMM | 0 | 1 | P | RR | RIS |

| RR | RIS | |
|---|---|---|
| 1 | 0 | IRRリード |
| 1 | 1 | ISRリード |

1：ポールコマンド
0：ノーポールコマンド

| ESMM | SMM | |
|---|---|---|
| 1 | 0 | スペシャルマスクをリセット |
| 1 | 1 | スペシャルマスクをセット |

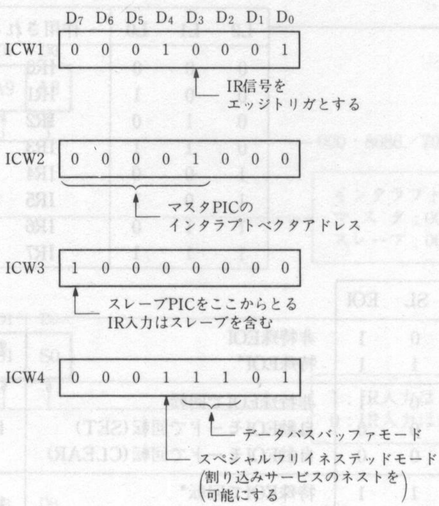
### ●ポールコマンドに対する読み込みワード



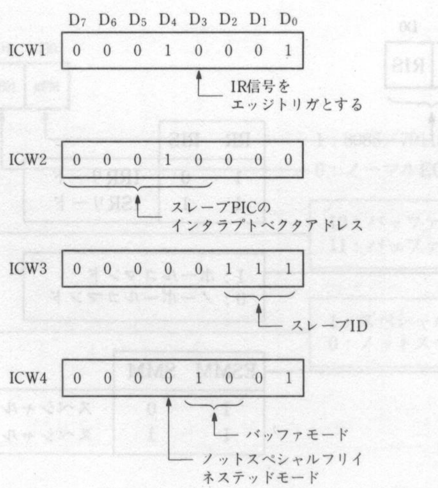

## 1.3　初期化プログラム

PC-98XL/XA では μPD8259A に対して次のような初期化を行う.

### (1) μ2D8259A マスタ初期化




### (2) μPD8259A スレーブの初期化



## 1.4　PIC 制御のプログラム例

(1) ISR の読み込み(割り込みサービス中の割り込みレベルを確認する)

(2) IMR の読み込み／書き込み(PIC の割り込みマスク状態の確認，設定)

(3) EOI の送り出し(割り込み終了を通知し，割り込みサービス状態をリセットする)

(4) IRR の読み込み(割り込み要求レジスタを読み，要求している割り込みレベルを確認する)
・割り込み禁止状態(IF＝0)中のプログラムから割り込み要求があるかどうかをチェックする.
・割り込みサービスの制御をソフト的に行う.
(IRR を読んで，割り込みベクタを INT でコールするなど)

(5) ハードウェア割り込みの許可，保留

ハードウェア割り込みの許可，保留は，CPU レベルで制御される．これは割り込み許可フラグ(IF)の状態によって決定される.

① IF＝0(割り込み許可フラグをオフにしている)の場合

CPU は PIC からの INT 信号を無視し，次の命令を実行する.

PIC は CPU からの応答があるか，要求の必要がなくなるまで，INT 信号をアクティブの状態のままにしておく.

CLI(Clear Interrupt-enable flag)の命令により IF＝0 となる.

② IF＝1(割り込み許可フラグをオンにしている)の場合

CPU は割り込みを認め，それを処理する.

CPU は割り込みを受付けるために $\overline{\text{INTA}}$ バスサイクルを 2 つ続けて実行する．最初の $\overline{\text{INTA}}$ で PIC に応答し，PIC はこれに応えてサービスを要求した機器の割り込みレベルを知らせる．0～255の 1 バイトデータが PIC から CPU に送られる.

CPU はこれによって，応答する割り込みベクタテーブルのエントリを探し，現在の CS, IP, フラグをスタックエリアに格納し，探し出した CS, IP をそれぞれセットする．セットされた状態で実行をはじめる.

STI(Set interrupt-enable flag)命令で IF＝1 となる.

注意：割り込みベクタの内容を変更するとか，現在の割り込みサービスプログラムを急いで処理するとか，ハードウェア割り込みに入られたくない状態がある．この場合は CLI 命令を実行し，できるだけ短時間の割り込み禁止状態にとどめ，必要がなくなれば STI 命令を実行し，割り込み許可状態にすること.

## 1.5　NMI(Non Maskable Interrupt)F/F

| 命　令 | I/Oポート アドレス | R/W |
|---|---|---|
| セット | 52 | W |
| リセット | 50 | W |

本F/Fは〝1〞の時，RAMのパリティエラーが起きた場合，NMIを発生させる．本F/Fが〝0〞ならば，上記パリティエラーが起こってもCPUには割り込みが発生しない．

なお本F/Fとは無関係に，システムポートBのD1，D2ビットを読み出すことにより，標準実装RAMおよび拡張RAMのパリティエラーが起こったかどうかを判別することが出来る(ただし，システムポートCのD4ビットが〝0〞である時は，パリティエラーは常に無視され，ポートBは常にエラーなしの状態となり，NMIも発生しない)．

# 第2章

# DMAコントローラ

## 2.1 DMAコントローラ μPD8237AC

DMAコントローラ μPD8237AC-5は，CPUを介さずに，周辺装置とメモリの間のデータ転送を直接行うためのものである．

・チャンネル割り当て

| チャネル | PC-98XL ノーマルモード | PC-98XL ハイレゾリューションモード | PC-98XA | 優先順位 |
|---|---|---|---|---|
| 0 | 固定ディスク | 固定ディスク | 未使用 | 高 |
| 1 | 未使用 | 1MB FD | 1MB FD | |
| 2 | 1MB FD | 未使用 | 640KB FD | |
| 3 | 640KB FD | 640KB FD | 固定ディスク | 低 |

上表で示すように，モードにより未使用チャンネルが異なるためDMAを使用するオプションボードは，チャンネル切り替え可能に設計する必要がある（PC-98XLの拡張スロットにはチャネル0および3のみ出力されている）．

・データ転送幅

8ビット/16ビット．

・アドレス範囲

メモリを直接アドレスできる範囲は64Kバイト．

DMAバンクアドレスレジスタの制御により24ビットアドレス可能．

レングスカウンタとして μPD8237A内のものを使用するため，64Kバイト境界をまたぐ転送は，バンクレジスタをオートインクリメントモードに設定することにより行う．ただしPC-98XAでは1Mバイト境界をまたぐ転送はできない．

・カスケード接続可能（チャンネル0以外）

・転送モード　シングルモード

・G-VRAM，T-VRAMのDMA転送は保証されない．

・G-VRAMのグラフィックチャージャモードでのDMAは不可能．

・PC-98XL ノーマルモード時のアドレス

16Mバイトのメモリ空間を指定する24ビットのアドレスは，上位8ビットをバンクレジスタにセットし，下位16ビットをDMAコントローラ8237にセットしなければならない．8237内の16ビットは，アドレスを加減算する機能があり，また上位4ビットのバンクレジスタは加算を選択できる．したがって，メモリの64Kバイト境界をまたぐDMA転送を行うことも可能である．

通常ノーマルモード時は，アドレスの最上位4ビットは0にマスクされる(1MBのアドレス範囲)．

・PC-98XL ハイレゾリューションモード時のアドレス

DMAアドレスは，16Mバイトのメモリ空間を指定するために24ビットの幅を持っている．8237にはアドレスの下位16ビットをセットし，アドレスの最上位8ビットは，バンクレジスタにセットする．

バンクレジスタは，8237内のアドレスと連結して内容を加算する機能を持っているため，64Kバイト境界をまたぐDMA転送を行うことができる．

16ビット転送の時，8237のアドレスレジスタの下位15ビットが，アドレスバスの1～15ビットに出力され(ワードアドレス)，アドレスレジスタの最上位ビットは無視される．

・$\mu$PD8237Aはレジスタの形で344ビットの内部メモリを持っている．

| レジスタ名称 | 大きさ | 個数 |
|---|---|---|
| ベースアドレスレジスタ | 16ビット | 4 |
| ベースワードカウントレジスタ | 16ビット | 4 |
| カレントアドレスレジスタ | 16ビット | 4 |
| カレントワードカウントレジスタ | 16ビット | 4 |
| テンポラリアドレスレジスタ | 16ビット | 1 |
| テンポラリワードカウントレジスタ | 16ビット | 1 |
| ステータスレジスタ | 8ビット | 1 |
| コマンドレジスタ | 8ビット | 1 |
| テンポラリレジスタ | 8ビット | 1 |
| モードレジスタ | 6ビット | 4 |
| マスクレジスタ | 4ビット | 1 |
| リクエストレジスタ | 4ビット | 1 |

## 2.2　I/Oアドレスと命令

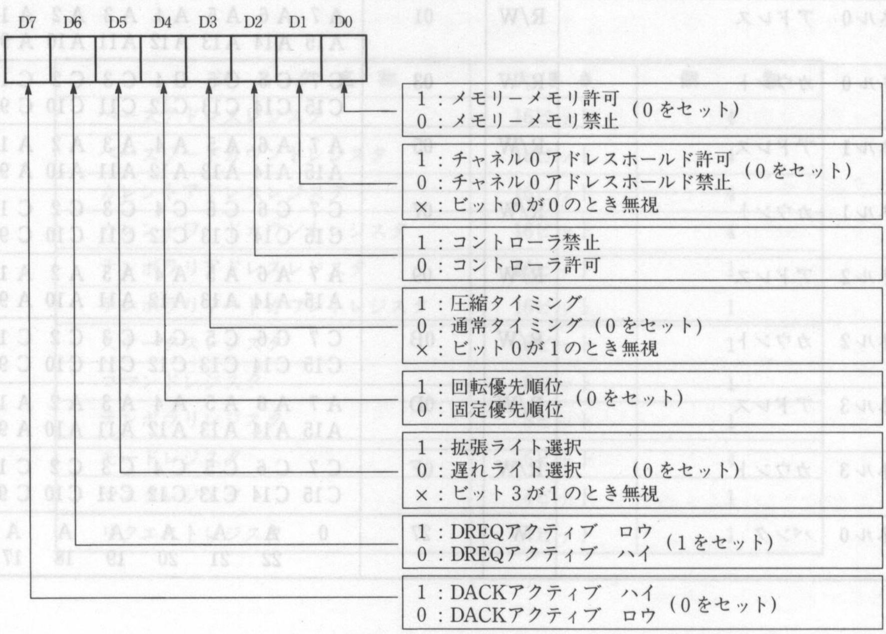
### (1) 命令一覧

| 命　　　令 | READ<br>/WRITE | I/Oポート<br>アドレス | データ<br>$D_7$ | <br>$D_6$ | <br>$D_5$ | <br>$D_4$ | <br>$D_3$ | <br>$D_2$ | <br>$D_1$ | <br>$D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|
| ライト　コマンド | W | 11 | K<br>S | D<br>S | W<br>S | P<br>R | T<br>M | C<br>E | A<br>H | M<br>M |
| ライト　モード | W | 17 | M<br>S<br>1 | M<br>S<br>0 | I<br>D | A<br>T | T<br>R<br>1 | T<br>R<br>0 | C<br>S<br>1 | C<br>S<br>0 |
| ライト　リクエスト | W | 13 | — | — | — | — | — | R<br>B | C<br>S<br>1 | C<br>S<br>0 |
| ライトシングルマスクレジスタビット | W | 15 | — | — | — | — | — | M<br>K | C<br>S<br>1 | C<br>S<br>0 |
| ライトオールマスクレジスタビット | W | 1F | — | — | — | — | M<br>B<br>3 | M<br>B<br>2 | M<br>B<br>1 | M<br>B<br>0 |
| リードステータス | R | 11 | R<br>Q<br>3 | R<br>Q<br>2 | R<br>Q<br>1 | R<br>Q<br>0 | T<br>C<br>3 | T<br>C<br>2 | T<br>C<br>1 | T<br>C<br>0 |
| チャネル0　アドレス | R/W | 01 | A 7<br>A15 | A 6<br>A14 | A 5<br>A13 | A 4<br>A12 | A 3<br>A11 | A 2<br>A10 | A 1<br>A 9 | A 0<br>A 8 |
| チャネル0　カウント | R/W | 03 | C 7<br>C15 | C 6<br>C14 | C 5<br>C13 | C 4<br>C12 | C 3<br>C11 | C 2<br>C10 | C 1<br>C 9 | C 0<br>C 8 |
| チャネル1　アドレス | R/W | 05 | A 7<br>A15 | A 6<br>A14 | A 5<br>A13 | A 4<br>A12 | A 3<br>A11 | A 2<br>A10 | A 1<br>A 9 | A 0<br>A 8 |
| チャネル1　カウント | R/W | 07 | C 7<br>C15 | C 6<br>C14 | C 5<br>C13 | C 4<br>C12 | C 3<br>C11 | C 2<br>C10 | C 1<br>C 9 | C 0<br>C 8 |
| チャネル2　アドレス | R/W | 09 | A 7<br>A15 | A 6<br>A14 | A 5<br>A13 | A 4<br>A12 | A 3<br>A11 | A 2<br>A10 | A 1<br>A 9 | A 0<br>A 8 |
| チャネル2　カウント | R/W | 0B | C 7<br>C15 | C 6<br>C14 | C 5<br>C13 | C 4<br>C12 | C 3<br>C11 | C 2<br>C10 | C 1<br>C 9 | C 0<br>C 8 |
| チャネル3　アドレス | R/W | 0D | A 7<br>A15 | A 6<br>A14 | A 5<br>A13 | A 4<br>A12 | A 3<br>A11 | A 2<br>A10 | A 1<br>A 9 | A 0<br>A 8 |
| チャネル3　カウント | R/W | 0F | C 7<br>C15 | C 6<br>C14 | C 5<br>C13 | C 4<br>C12 | C 3<br>C11 | C 2<br>C10 | C 1<br>C 9 | C 0<br>C 8 |
| チャネル0　バンク | W | 27 | 0 | A<br>22 | A<br>21 | A<br>20 | A<br>19 | A<br>18 | A<br>17 | A<br>16 |

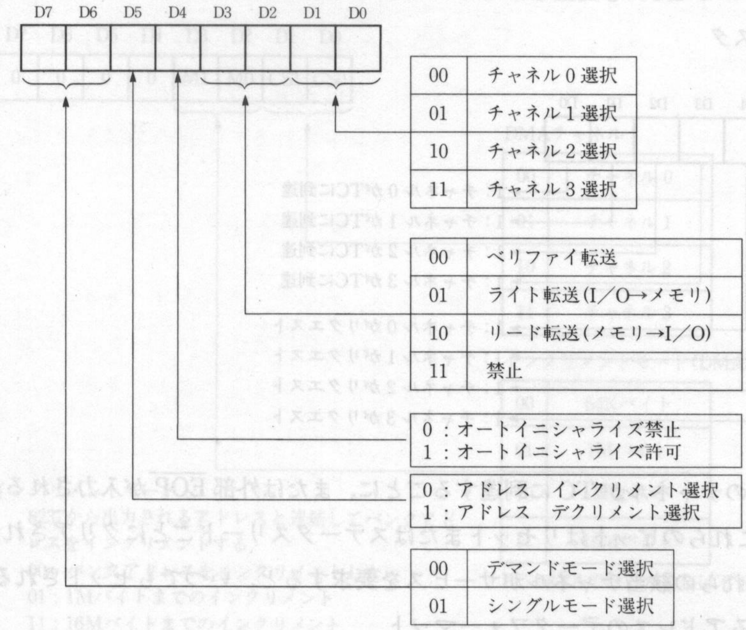
| 命令 | READ /WRITE | I/O ポート アドレス | データ D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| チャネル1 バンク | W | 21 | 0 | A22 | A21 | A20 | A19 | A18 | A17 | A16 |
| チャネル2 バンク | W | 23 | 0 | A22 | A21 | A20 | A19 | A18 | A17 | A16 |
| チャネル3 バンク | W | 25 | 0 | A22 | A21 | A20 | A19 | A18 | A17 | A16 |
| クリアマスクレジスタ | W | 1D | — | — | — | — | — | MK | CS1 | CS0 |
| リードテンポラリレジスタ | R | 1B | — | — | — | — | — | — | — | — |
| クリアバイトポインタフリップフロップ | W | 19 | — | — | — | — | — | — | — | — |
| マスタ クリア | W | 1B | — | — | — | — | — | — | — | — |
| バンクアドレスオートインクリメントモードレジスタ | W | 29 | 0 | 0 | 0 | 0 | M1 | M0 | CS1 | CS0 |

## (2) レジスタの説明

### ① コマンドレジスタ(カッコ内はシステム設定値を示す)



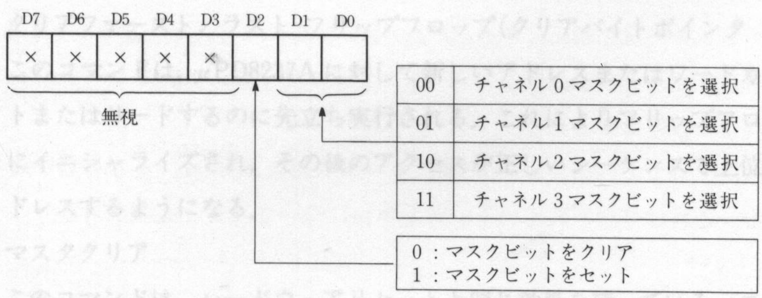
② モードレジスタ



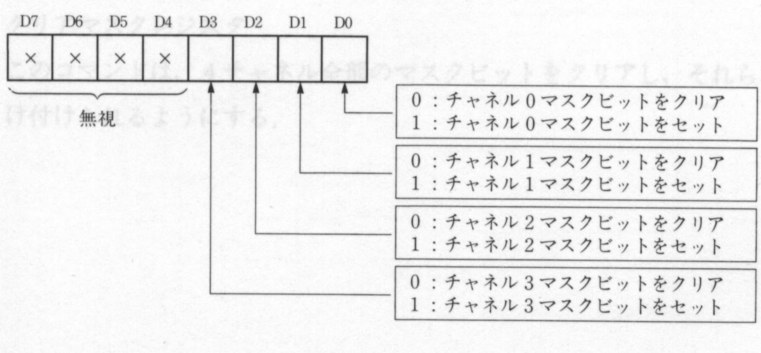

③ リクエストレジスタ使用禁止

④ マスクレジスタ

a) シングルマスクレジスタビット

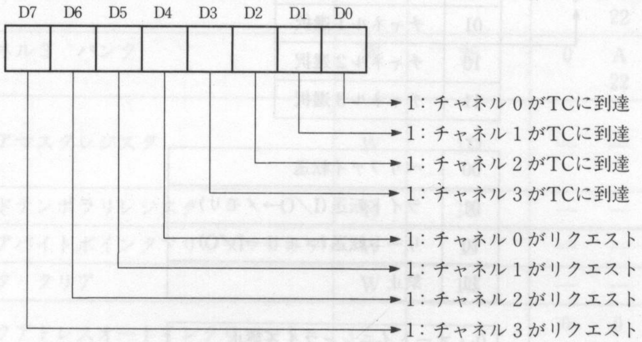



b) オールマスクレジスタビット



各マスクビットは，その付随するチャネルがオートイニシャライズにプログラムされていなければ，そのチャネルが$\overline{\text{EOP}}$を発生したときにセットされる．

⑤ステータス　レジスタ




ビット0－3は，そのチャネルがTCに到達するごとに，または外部$\overline{\text{EOP}}$が入力されるごとにセットされる．これらのビットはリセットまたはステータスリードごとにクリアされる．

ビット4－7は，それらの該当チャネルがサービスを要求すると，いつでもセットされる．

⑥バンクにセットするアドレスのデータフォーマット

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| × | A22 | A21 | A20 | A19 | A18 | A17 | A16 |

バンクへの書き込みはできるが，バンクの内容の読み出しコマンドはない．

⑦バンクモード(バンクアドレスオートインクリメントモードレジスタ)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | M1 | M0 | CS1 | CS0 |

DMAチャネル

| | |
|---|---|
| 00 | チャネル0 |
| 01 | チャネル1 |
| 10 | チャネル2 |
| 11 | チャネル3 |

インクリメントモード(DMA境界)

| | |
|---|---|
| 00 | 64Kバイト |
| 01 | 1Mバイト |
| 10 | 設定不可 |
| 11 | 16Mバイト |

PC-98XAでは11の設定は不可

インクリメントモード：
8237から出力されるアドレスと連結してバンクアドレスをインクリメントする．
00：バンクアドレスをインクリメントしない
01：1Mバイトまでのインクリメント
11：16Mバイトまでのインクリメント

注意：PC-98XLの70116モードの場合，データの上位4ビットはマスクされて0となり，1Mバイト以上の空間にDMA転送することはできない．ただし，DMA境界を16Mバイトに設定した場合，アドレスカウントアップにより1Mバイトを越える場合があるため注意すること．

⑧ソフトウェアコマンド

a）クリアファースト／ラスト フリップフロップ(クリアバイトポインタ フリップフロップ)

このコマンドは，μPD8237Aに対して新しいアドレスまたはワードカウント情報をライトまたはリードするのに先立ち実行される．これによりフリップフロップが既知の状態にイニシャライズされ，その後のアクセスが正しいシーケンスで上位と下位バイトをアドレスするようになる．

b）マスタクリア

このコマンドは，ハードウェアリセットと同じ効果を持っている．コマンド，ステータス，リクエスト，テンポラリ，そして内部ファースト／ラスト フリップフロップがクリアされ，マスクレジスタがセットされる．μPD8237Aはアイドルサイクルに入る．

c）クリアマスクレジスタ

このコマンドは，4チャネル全部のマスクビットをクリアし，それらがDMA要求を受け付けられるようにする．

### (3) その他

① 転送モード

PC-98XL/XAでは，シングルトランスファモードのみを使用している．すなわち，デバイスは1回の転送だけを行うようにプログラムされる．各転送後，ワードカウントがデクリメントされ，アドレスがデクリメントまたはインクリメントされる．1回のDMAリクエストに対して1バイトの転送を行う．

② オートイニシャライズ

モードレジスタのビットをプログラムする事により，あるチャネルをオートイニシャライズチャネルとして設定できる．オートイニシャライズに初期設定がされていると，$\overline{\text{EOP}}$に続いて，カレントアドレスとカレントワードカウントレジスタのもとの値が，そのチャネルのベースアドレスとベースワードカウントレジスタから自動的にリトスアされる．各ベースレジスタには，各カレントレジスタの内容が自動的にロードされ，DMAサービス中保持されている．チャネルがオートイニシャライズに設定されているときは，マスクビットはセットされない．そのためオートイニシャライズの後，有効なDREQが検出されるとすぐ，そのチャネルはCPUの仲介なしにサービスを実行できる．

ただし，DMAバンクアドレスレジスタはオートイニシャライズされないので，オートインクリメントモード時は注意すること．

③ プログラミングの注意点

μPD8237AC-5のプログラム中は，DMA要求が起きるのを避ける必要がある．このためにプログラム前にコントローラを禁止にする(コマンドレジスタのビット2をセットする)か，またはそのチャネルをマスクする必要がある．プログラム完了後，コントローラをイネーブルし，マスクされない状態にすること．

電源投入後には，すべての内部レジスタ(特にモードレジスタ)には，ある有効な値をロードする必要がある．これは使用されていないチャネルについても行う必要がある．

# 第3章
# タイマ

## 3.1 インターバルタイマ μPD8253C

・３組の16ビットカウンタにより構成される．

・カウントレートは 8MHz モードの場合 500.8ns(1.9968MHz)，PC-98XL 10MHz モードの場合 406.9ns(2.4576MHz)を使用する．

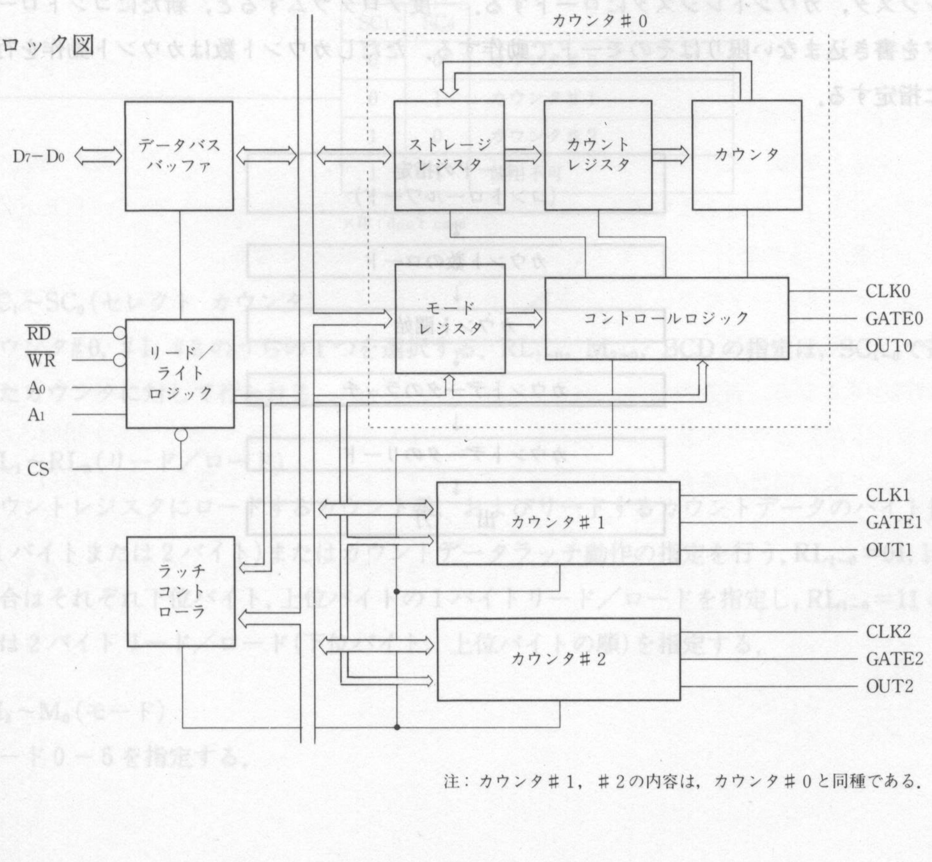
・３種類の動作モード(モード０，モード２，モード３)を使用する．

・それぞれのカウンタは次のように使用する．

・PC-98XL ノーマルモード時は PC-9801VM の資料を参照のこと．

| | |
|---|---|
| カウンタ０ | インターバルタイマ |
| カウンタ１ | スピーカー周波数設定 |
| カウンタ２ | RS-232C |

・ブロック図




注：カウンタ＃1，＃2の内容は，カウンタ＃0と同種である．

## 3.2 I/Oアドレスと命令

● PC-9801/E/F/M

| 命　令 | READ /WRITE | I/Oポート アドレス | データ $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|
| カウンタ0へのロード | W | 71 | $C_7$<br>$C_{15}$ | — | — | — | — | — | — | $C_0$<br>$C_8$ |
| カウンタ0をリード | R | 71 | $C_7$<br>$C_{15}$ | — | — | — | — | — | — | $C_0$<br>$C_8$ |
| カウンタ1へのロード | W | 73 | $C_7$<br>$C_{15}$ | — | — | — | — | — | — | $C_0$<br>$C_8$ |
| カウンタ1をリード | R | 73 | $C_7$<br>$C_{15}$ | — | — | — | — | — | — | $C_0$<br>$C_8$ |
| カウンタ2へのロード | W | 75 | $C_7$<br>$C_{15}$ | — | — | — | — | — | — | $C_0$<br>$C_8$ |
| カウンタ2をリード | R | 75 | $C_7$<br>$C_{15}$ | — | — | — | — | — | — | $C_0$<br>$C_8$ |
| モード　指定 | W | 77 | SC1 | SC0 | RL1 | RL0 | M2 | M1 | M0 | BCD |

カウンタを動作させる場合は，OUT命令によりコントロールワード，カウント数をあらかじめ指定する必要がある．プログラムはカウンタ別に行い，それぞれのカウンタに付属するモードレジスタ，カウントレジスタにロードする．一度プログラムすると，新たにコントロールワードを書き込まない限りはそのモードで動作する．ただしカウント数はカウント動作を行うごとに指定する．

モードの指定
（コントロールワード）
↓
カウント数のロード
↓
カウント開始
↓
カウントデータのラッチ
↓
カウントデータのリード
↓
出　力

### (1) コントロールワード

カウンタの動作モードを指定するためのもので，カウンタのモード指定およびカウントデータのラッチ動作に使用する．

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| $SC_1$ | $SC_0$ | $RL_1$ | $RL_0$ | $M_2$ | $M_1$ | $M_0$ | BCD |

| BCD | |
|---|---|
| 0 | バイナリカウント(16桁) |
| 1 | BCDカウント(4桁) |

(0をセット)

| $M_2$ | $M_1$ | $M_0$ | |
|---|---|---|---|
| 0 | 0 | 0 | モード0 |
| × | 1 | 0 | モード2 |
| × | 1 | 1 | モード3 |

(モード1, モード4, モード5は省略)

| $RL_1$ | $RL_0$ | |
|---|---|---|
| 0 | 0 | カウントラッチ動作 |
| 0 | 1 | LSBのリード／ロード |
| 1 | 0 | MSBのリード／ロード |
| 1 | 1 | LSB, MSBの順にリード／ロード |

| $SC_1$ | $SC_0$ | |
|---|---|---|
| 0 | 0 | カウンタ＃0 |
| 0 | 1 | カウンタ＃1 |
| 1 | 0 | カウンタ＃2 |
| 1 | 1 | 使用不可 |

×印：don't care

① $SC_1 \sim SC_0$(セレクト　カウンタ)

カウンタ#0，#1，#2のうちの1つを選択する．$RL_{1\sim0}$，$M_{2\sim0}$，BCDの指定は，$SC_{1\sim0}$で選択したカウンタに対して行われる．

② $RL_1 \sim RL_0$(リード／ロード)

カウントレジスタにロードするカウント数，およびリードするカウントデータのバイト長，(1バイトまたは2バイト)またはカウントデータラッチ動作の指定を行う．$RL_{1\sim0}$＝01, 10の場合はそれぞれ下位バイト, 上位バイトの1バイトリード／ロードを指定し, $RL_{1\sim0}$＝11の場合は2バイトリード／ロード(下位バイト，上位バイトの順)を指定する．

③ $M_2 \sim M_0$(モード)

モード0～5を指定する．

④ BCD

バイナリカウントおよびBCDカウントの選択を行う.

## (2) モードの指定

① モード0：カウント終了時の割り込み

指定したカウントが終了するとOUTが1になる.

コントロールワードによりモードを指定するとOUTは0となり，カウント数をロードすると直ちにカウントが開始される. カウントが終了するとOUTは0から1となる. カウント終了後は，新しいカウント数をロードするまでOUTは1の状態を続ける. なお，OUTが1となった後もカウンタのデクリメントは続けられる.

カウント中に新たなカウント数をロードすると，新しいカウント数でカウントを開始する.

カウント数が2バイトの場合は，下位バイトをロードするとカウントが停止し，上位バイトをロードすると新しいカウントが開始される.

② モード2：レートジェネレータ

入力クロックのn分周カウンタである. ただし，OUTは1クロックサイクルだけ0となるので，デューティー比は1/nとなる.

コントロールワードによりモード2を指定するとOUT=1となる. カウント数をロードするとカウントを開始し，カウントの最後で1クロックサイクルの期間OUT=0となる.

カウント中に新たなカウント数をロードしても，実行中のカウントには影響を与えないが，次のサイクルからは新しいカウント数でカウントが行われる.

③ モード3：方形波レートジェネレータ

モード2と同様，入力クロックのn分周カウンタである. ただし，カウント数が偶数の場合のデューティー比は1/2である. 奇数の場合のデューティー比はn−1/2nで，たとえば，カウント数n=5の時はデューティー比2/5(アクティブロウ)となる.

コントロールワードでモード3を選択するとOUT=1となり，カウント数をロードするとカウントが開始される. カウント数が偶数の時は，カウントの前半1/2がOUT=1，後半1/2がOUT=0となる. 奇数の時は，前半n+1/2がOUT=1，後半n−1/2がOUT=0となる.

カウント中にカウント数をロードすると，次のサイクルから新しいカウントが開始される.

カウント数が偶数の場合はカウンタは2ずつデクリメントされ，奇数の場合はOUT=1のときには最初の1クロックで1デクリメントされ，2クロック目からは2ずつデクリメントされる. OUT=0のときには最初の1クロックで3デクリメントされ，2クロック目からは2ずつデクリメントされる.

### (3) 動作に関する注意点

① 書き込み動作

カウント数の書き込みはコントロールワード書き込みの直後に続いて行う必要はなく，コントロールワード書き込み以後，どの時点でも行う事ができる．ただし2バイトを指定した場合は下位，上位バイトは切り離せないので，必ず下位バイトの直後に続いて上位バイトを書き込む事が必要である．

② カウント動作

カウント数の指定は1バイトのみとすることができるが，カウンタは常に2バイトで動作するため，$RL_{1\sim0}$で下位バイトを指定すると上位バイトの初期値は0，上位バイトを指定すると下位バイトの初期値は0となる．

③ 読み出し動作

カウント中のカウンタの内容を読み出す場合は，カウンタの内容を一時ストレージレジスタにロードし，それからリード動作を用いてラッチされているデータを読み出す．

ラッチは，読み出したい時点に次のようなコントロールワードを書き込むことにより行う．

$$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0 = \underbrace{SC1, SC0,}_{\text{カウンタの選択}}\ 0,\ 0,\ \underbrace{X,\ X,\ X,\ X}_{\text{不定}}$$

コントロールワードの書き込みにより，$SC_{1\sim0}$で選択されたカウンタの内容がストレージレジスタにラッチされる．この後カウンタを選択してストレージレジスタにラッチされているデータを読み出す．

## 3.3　タイマ設定値

### (1) #0インターバルタイマ

割り込みインターバル n ミリ秒に対して，下記に近い値をセットする．

・10MHz モードの場合

n×2457.6 に近い整数(最大26.666ミリ秒まで可)

・8MHz モードの場合

n×1996.8 に近い整数(最大32.821ミリ秒まで可)

### (2) #1スピーカー周波数発生

・方形波モードでスピーカー周波数を発生する．

・設定値 n＝3F0H(2KHz)

### (3) #3 RS-232C

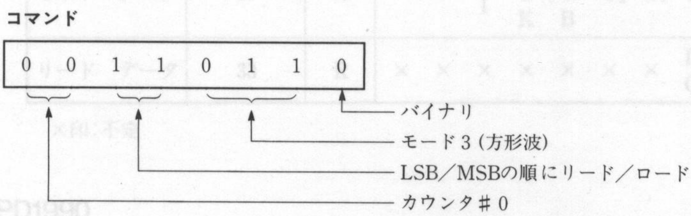
| 通信速度(ボー) | 同　期　式 | | 調歩同期式　1/16 | | 調歩同期式　1/64 | |
|---|---|---|---|---|---|---|
| | 10MHz | 8MHz | 10MHz | 8MHz | 10MHz | 8MHz |
| 9600 | 256 | 208 | 16 | 13 | 使用不可 | 使用不可 |
| 4800 | 512 | 416 | 32 | 26 | 8 | 使用不可 |
| 2400 | 1024 | 832 | 64 | 52 | 16 | 13 |
| 1200 | 2048 | 1664 | 128 | 104 | 32 | 26 |
| 600 | 4096 | 3328 | 256 | 208 | 64 | 52 |
| 300 | 8192 | 6656 | 512 | 416 | 128 | 104 |
| 150 | 16384 | 13312 | 1024 | 832 | 256 | 208 |
| 75 | 32768 | 26624 | 2048 | 1664 | 512 | 416 |

| 通信速度 | 同期刻時式 |
|---|---|
| 1200 | 104 |

## 3.4　タイマ利用のプログラム例

### (1) タイマのセット

```
        ∫
MOV   AL, 36H  ………… コマンドセット
OUT   77H, AL
MOV   AL, 00H
OUT   71H, AL  …………LSB のセット
MOV   AL, 4EH
OUT   71H, AL  …………MSB のセット
IN    AL, 02H  ………… 割り込みマスクレジスタのリード
AND   AL, 0FEH………… 割り込みマスクレジスタのセット
OUT   02H, AL  ………… 割り込みマスクレジスタのライト
        ∫
```




LSB

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

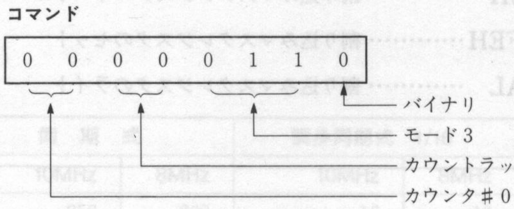
MSB

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

10msec×1996.8＝19968
＝4E00H (8MHzモード)

### (2) タイマのリード

```
        ∫
MOV   AL, 06H
OUT   77H, AL   …… カウンタの内容を読み出すためにストレージレジスタでラッチ
IN    AL, 71H   ┐
MOV   BL, AL    ├ LSB, MSB の順にリード(BX にタイマ値が入る)
IN    AL, 71H   ┘
MOV   BH, AL
        ∫
```



# 第4章

# カレンダ時計

カレンダ時計には，時刻およびカレンダのデータをシリアルに入出力する $\mu$PD4990A（PC-98XA では $\mu$PD1990C）シリアルI/Oカレンダ時計が使用されており，時，分，秒および月，日，曜日を扱うことができる．また，$\mu$PD4990A の拡張モードを利用することにより，年を扱うこともできる．データは秒，分，時，日，曜日，月の順に読み出される．また月の大小は自動的に判別する．カレンダ時計はバッテリーによってバックアップされているため，電源をOFFしても約2ケ月間は動作し続けることができる．

## 4.1 I/Oアドレスと命令

| 命令 | I/Oポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|
| セットレジスタ | 20 | W | × | × | DI | CLK | STB | $C_2$ | $C_1$ | $C_0$ |
| リード　データ | 33 | R | × | × | × | × | × | × | × | DO |

×印：不定

### 4.1.1 $\mu$PD1990

#### (1) セットレジスタ

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| × | × | DI | CLK | STB | $C_2$ | $C_1$ | $C_0$ |

$C_2$～$C_0$：コマンドコード
STBと共に用いて $\mu$PD1990 へパラレルコマンドを与える．

| $C_2$ | $C_1$ | $C_0$ | 機能 |
|---|---|---|---|
| 0 | 0 | 0 | レジスタ ホールド |
| 0 | 0 | 1 | レジスタ シフト |
| 0 | 1 | 0 | タイムセット＆カウンタホールド |
| 0 | 1 | 1 | タイム リード |
| 1 | 0 | 0 | 禁止 |
| 1 | 0 | 1 | 禁止 |
| 1 | 1 | 0 | 禁止 |
| 1 | 1 | 1 | 禁止 |

STB：ストローブ
$\mu$PD1990 へコマンドをセットするタイミングを与える．

CLK：クロック
DOと共に用いて，$\mu$PD1990 へ時刻をセットしたり，時刻を読み出すために用いる．

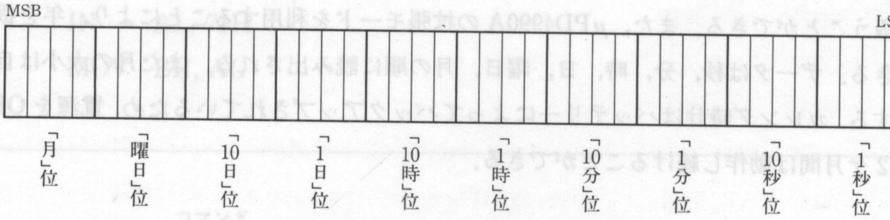
DI：入力データ（1ビット毎に入力される）
CLKと共に用いて $\mu$PD1990 へ時刻データをセットする．

### (2) リードデータ

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| × | × | × | × | × | × | × | DO |

D0 → 出力データ(1ビット毎に出力される)
CLKと共に用いてμPD1990から時刻を読み出す.

### (3) 入出力データ形式




入出力は常にLSBから1ビットずつ行われる.

データ形式はBCDであるが,「月」のみHEXA-DECIMALである.

曜日は,次のように示される.

MSB ─┐　┌─ LSB

0 0 0 0 ― 日曜
0 0 0 1 ― 月曜
0 0 1 0 ― 火曜
0 0 1 1 ― 水曜
0 1 0 0 ― 木曜
0 1 0 1 ― 金曜
0 1 1 0 ― 土曜

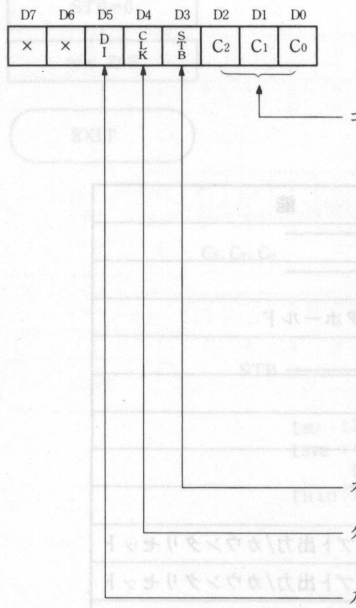
| 項目 | 形　式 | 範　囲 |
|---|---|---|
| 月 | HEX DECIMAL | 01H～0CH |
| 曜 | BCD | 00H～06H |
| 日 | BCD | 01H～31H |
| 時 | BCD | 00H～23H |
| 分 | BCD | 00H～59H |
| 秒 | BCD | 00H～59H |

## 4.1.2　μPD4990A

μPD4990Aは μPD1990の上位互換のLSIであり，テストモード(PC-98XAでは使用せず)を除き μPD1990とコンパチブルな動作が可能である．また μPD4990Aには年を扱う事のできる拡張モード(シリアルコマンドによる)が用意されており，拡張モード時の内部レジスタは52ビット(コマンド4ビット＋データ48ビット)となる．

シリアルコマンドの発行は，μPD1990のC2, C1, C0によるパラレル方式とは異なり，DIから4ビットのシリアル情報の形式をとる．シリアルコマンドは，その時点のモードには関わらず(レジスタホールドコマンドではシリアルコマンドレジスタのホールドは行わない)随時 μPD4990Aに書き込むことができ，C2, C1, C0＝1，1，1とした状態でストローブを入力することにより，コマンドとして有効になる．

### (1) セットレジスタ




コマンドコード
STBと共に用いて μPD4990へパラレルコマンドを与える．

| $C_2$ | $C_1$ | $C_0$ | 機　能 |
|---|---|---|---|
| 0 | 0 | 0 | レジスタ ホールド |
| 0 | 0 | 1 | レジスタ シフト |
| 0 | 1 | 0 | タイムセット&カウンタホールド |
| 0 | 1 | 1 | タイム リード |
| 1 | 0 | 0 | TP＝64Hzセット |
| 1 | 0 | 1 | TP＝256Hzセット |
| 1 | 1 | 0 | TP＝2048Hzセット |
| 1 | 1 | 1 | 拡張モードコマンド |

ストローブ
μPD4990へコマンドをセットするタイミングを与える．

クロック
DOと共に用いて，シリアルコマンドを μPD4990へ時刻をセットしたり，時刻を読み出すために用いる．

入力データ(1ビット毎に入力される)
CLKと共に用いて μPD4990へシリアルコマンドおよびへ時刻データをセットする

注：パラレルコマンドで日付，時刻を書き込むと，年のデータは壊れる．
その場合は日付，時刻の再設定を行うこと．

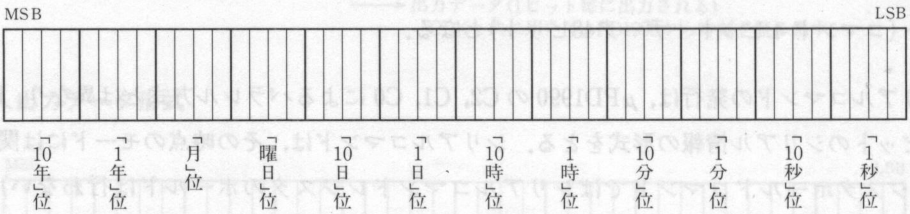
### (2) リードデータ

μPD1990に同じ．

### (3) 入出力データ形式

年データは西暦 2 桁である．

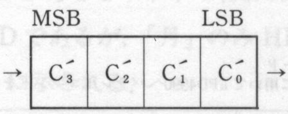
その他は μPD1990 に同じ．




### (4) シリアルコマンド

DI から 4 ビットのデータを送ることにより，コマンドを指定する．

年機能が有効．

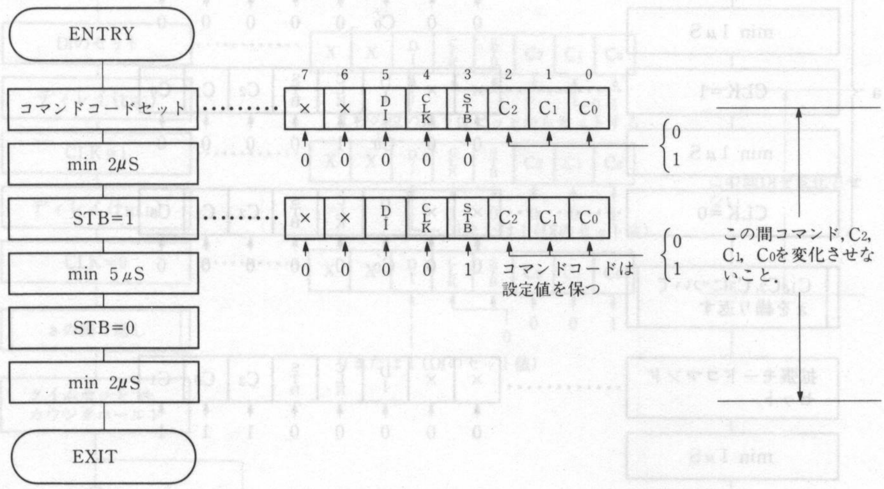


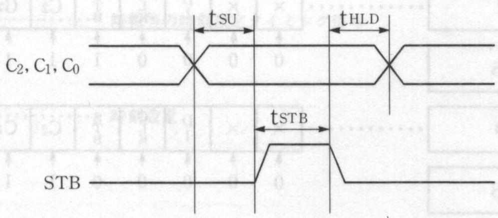

| $C_3'$ | $C_2'$ | $C_1'$ | $C_0'$ | 機　能 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | レジスタホールド |
| 0 | 0 | 0 | 1 | レジスタシフト |
| 0 | 0 | 1 | 0 | タイムセット＆カウンタホールド |
| 0 | 0 | 1 | 1 | タイムリード |
| 0 | 1 | 0 | 0 | TP＝64Hz |
| 0 | 1 | 0 | 1 | TP＝256Hz |
| 0 | 1 | 1 | 0 | TP＝2048Hz |
| 0 | 1 | 1 | 1 | TP＝4096Hz |
| 1 | 0 | 0 | 0 | TP＝1sec.　インタラプト出力/カウンタリセット |
| 1 | 0 | 0 | 1 | TP＝10sec.　インタラプト出力/カウンタリセット |
| 1 | 0 | 1 | 0 | TP＝30sec.　インタラプト出力/カウンタリセット |
| 1 | 0 | 1 | 1 | TP＝60sec.　インタラプト出力/カウンタリセット |
| 1 | 1 | 0 | 0 | インタラプト出力リセット |
| 1 | 1 | 0 | 1 | インタラプトタイマスタート |
| 1 | 1 | 1 | 0 | インタラプトタイマストップ |
| 1 | 1 | 1 | 1 | テストモードセット |

## 4.2 使用方法

### (1) コマンドのセット

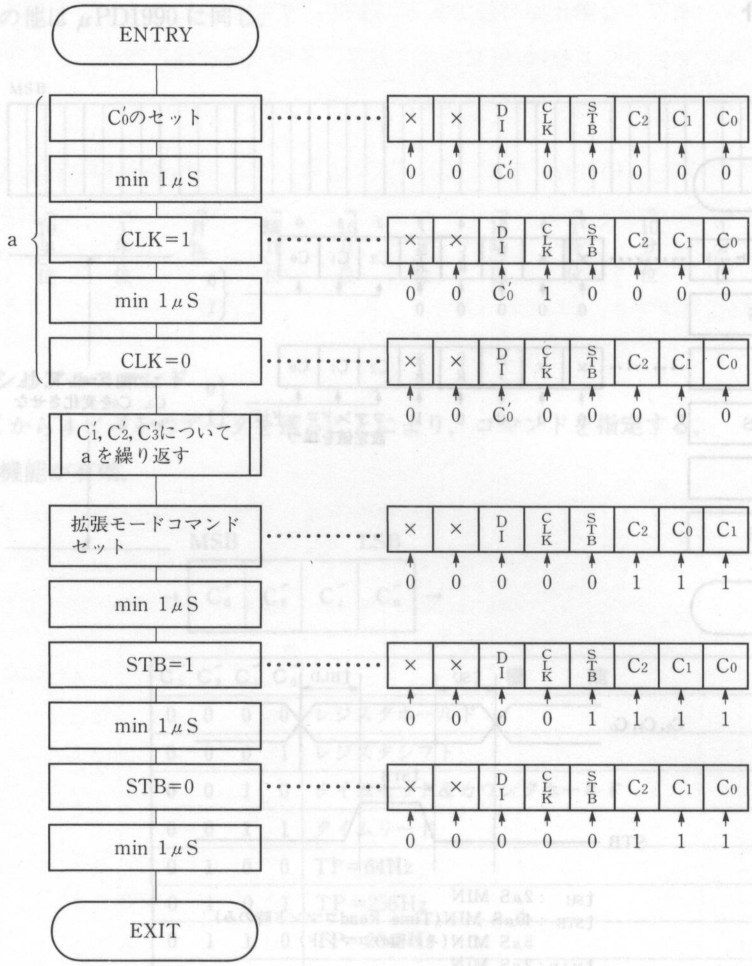
・パラレルコマンド



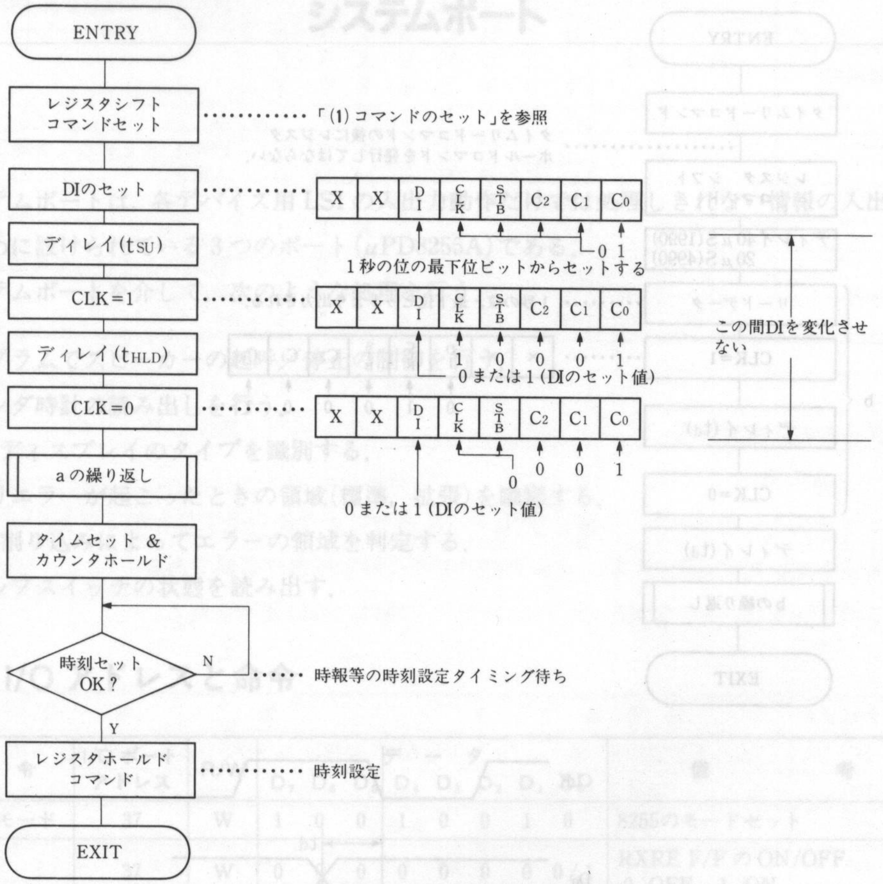



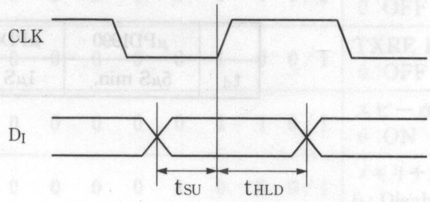

$t_{SU}$ : 2μS MIN
$t_{STB}$ : 40μS MIN(Time Readコマンド時のみ)
5μS MIN(その他のコマンド)
$t_{HLD}$ : 2μS MIN

・シリアルコマンド


ENTRY
C'0のセット
min 1μS
a
CLK=1
min 1μS
CLK=0
C'1, C2, C3について
aを繰り返す
拡張モードコマンド
セット
min 1μS
STB=1
min 1μS
STB=0
min 1μS
EXIT
× × DI CLK STB C2 C1 C0
0 0 C'0 0 0 0 0 0
× × DI CLK STB C2 C1 C0
0 0 C'0 1 0 0 0 0
× × DI CLK STB C2 C1 C0
0 0 C'0 0 0 0 0 0
× × DI CLK STB C2 C0 C1
0 0 0 0 0 1 1 1
× × DI CLK STB C2 C1 C0
0 0 0 0 1 1 1 1
× × DI CLK STB C2 C1 C0
0 0 0 0 0 1 1 1

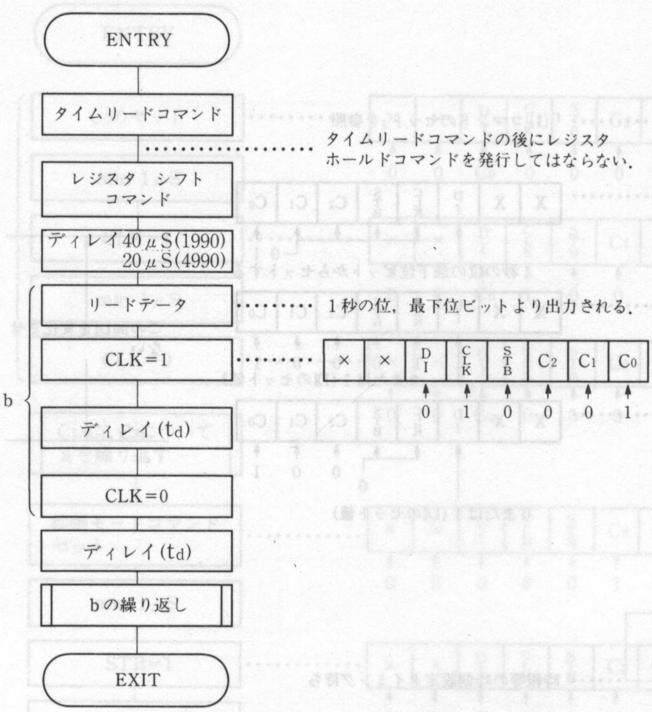
## (2) 時刻の設定

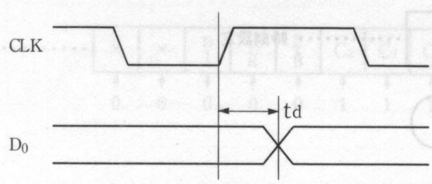



| | μPD1990 | μPD4990A |
|---|---|---|
| $t_{SU}$ | 2μS min. | 1μS min. |
| $t_{HLD}$ | 2μS min. | 1μS min. |

## (3) 時刻の読み出し




| | μPD1990 | μPD4990A |
|---|---|---|
| $t_d$ | 5μS min. | 1μS min. |

# 第5章

# システムポート

システムポートは，各デバイス用LSIの入出力動作だけでは処理しきれない情報の入出力を行うために設けられている3つのポート($\mu$PD8255A)である．

システムポートを介して，次のような処理を行う．

・プログラムでスピーカーの起呼／停止の制御を行う．
・カレンダ時計の読み出しを行う．
・CRTディスプレイのタイプを識別する．
・メモリエラーが起こったときの領域(標準，拡張)を識別する．
・NMI割り込みによってエラーの領域を判定する．
・ディップスイッチの状態を読み出す．

## 5.1 I/Oアドレスと命令

| 命令 | I/Oポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ライトモード | 37 | W | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 8255のモードセット |
| | 37 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | RXRE F/FのON/OFF<br>0:OFF　1:ON |
| | 37 | W | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0/1 | TXEE F/FのON/OFF<br>0:OFF　1:ON |
| | 37 | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0/1 | TXRE F/FのON/OFF<br>0:OFF　1:ON |
| ライトポートC | 37 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0/1 | スピーカ　F/FのON/OFF<br>0:ON　1:OFF |
| | 37 | W | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0/1 | メモリチェックEnableのON/OFF<br>0 : Disable, 1 : Enable |
| | 37 | W | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0/1 | SHUT1のON/OFF |
| | 37 | W | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0/1 | プリンタのPSTB信号マスクF/FのON/OFF<br>0:イネーブル，1:マスク |
| | 37 | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0/1 | SHUT0のON/OFF |
| ライトポートC | 35 | W | SHUT0 | PSTBM | SHUT1 | MCKEN | BUZ | TXRE | TXEE | RXRE | PORT Cの信号は本命令でもON/OFF可 |

| 命　令 | I/Oポート アドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| リードポートC（診断用） | 35 | R | SHUT0 | PSTBM | SHUT1 | MCKEN | BUZ | TXRE | TXEE | RXRE | PORT Cの状態を読み取る． |
| リードポートB | 33 | R | $\overline{\text{CI}}$ | $\overline{\text{CS}}$ | $\overline{\text{CD}}$ | INT3 | CRT | IMCK | EMCK | CDAT | PORT Bを通して各種信号を読み取る． |
| リードポートA | 31 | R | $\overline{\text{SW8}}$ | $\overline{\text{SW7}}$ | $\overline{\text{SW6}}$ | $\overline{\text{SW5}}$ | $\overline{\text{SW4}}$ | $\overline{\text{SW3}}$ | $\overline{\text{SW2}}$ | $\overline{\text{SW1}}$ | PORT Aを通して各種スイッチ信号を読み取る． |

## (1) ポートの入出力信号

① ポートA入力

ディップスイッチSW2の状態をCPUのIN命令により読み出すことができる．

スイッチオンで〝0〞，スイッチオフで〝1〞なので注意が必要である．

以下のデータは，PC-98XLのノーマルモード時にのみに使用される．

| データビット | 信号名 | 備　考 |
|---|---|---|
| $D^7$ | SW8 | ON：GDC 5MHz　　OFF：GDC 2.5MHz |
| $D^6$ | SW7 | 未使用 |
| $D^5$ | SW6 | 未使用 |
| $D^4$ | SW5 | ON：メモリスイッチ保持　　OFF：メモリスイッチ初期化 |
| $D^3$ | SW4 | ON：25行／画面　　OFF：20行／画面 |
| $D^2$ | SW3 | ON：80文字／行　　OFF：40文字／行 |
| $D^1$ | SW2 | ON：ターミナルモード　　OFF：BASIC |
| $D^0$ | SW1 | 常にOFF |

注：PC-98XA/XLハイレゾリューションモードではSW5のみ使用

② ポートB入力

ポートに入力される外部信号をCPUのIN命令によって読み出すことができる.

| データビット | 信号名 | 備考 |
|---|---|---|
| $D^7$ | $\overline{CI}$(RS-232C) | RS-232C　CI信号 |
| $D^6$ | $\overline{CS}$(RS-232C) | RS-232C　CS信号 |
| $D^5$ | $\overline{CD}$(RS-232C) | RS-232C　CD信号 |
| $D^4$ | INT3／SHUT0 | 固定ディスク　INT信号／SHUT0 |
| $D^3$ | CRT TYPE／SHUT1 | 1:高解像度, 0:標準解像度／SHUT1 |
| $D^2$ | 内部RAMパリティエラー | 標準RAMのパリティエラー |
| $D^1$ | 外部RAMパリティエラー | 拡張RAMのパリティエラー |
| $D^0$ | カレンダ時計の読み出しデータ | |

・INT3/SHUT0およびCRT TYPE/SHUT1の選択
PC-98XAではSHUT0, 1のみに固定されている.
PC-98XLのPower On/リセット後はSHUT0, 1が選択される. この時, ポートCは入力モードになっているため, 読みだしたときは必ず〝11〞となる. 以後, 8255のモードセットによりINT3・CRT TYPE/SHUT0・1のいずれかが選択される.

③ ポートC出力

CPUからのビットセット／リセット命令によってビット単位に制御される.

ポートCはリセット時(電源投入またはリセットキー押下)は入力モードになるので, 必ず各制御F/Fを初期設定する必要がある.

| データビット | 信号名 | 備考 |
|---|---|---|
| $D^7$ | SHUT0 | |
| $D^6$ | PSTBマスク | プリンタのPSTB信号のマスク |
| $D^5$ | SHUT1 | |
| $D^4$ | メモリチェックEnable | 1:エラー登録する(注1), 0:エラー登録しない |
| $D^3$ | ブザー制御F/F | 1:ブザー停止, 0:鳴動 |
| $D^2$ | TXR　Enable　F/F | RS-232CのTXRDYによる割り込みのEnable　(注2) |
| $D^1$ | TXE　Enable　F/F | RS-232CのTXEMPTYによる割り込みのEnable　(注2) |
| $D^0$ | RXR　Enable　F/F | RS-232CのRXRDYによる割り込みのEnable　(注2) |

注1: ポートBの$D^1$ビットに登録する.
注2: 1:対応した信号の割り込みを可能にする.
　　0:対応した信号が〝1〞になっても割り込みを発生させない.

| 命令 | I/Oポートアドレス | R/W | データ $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| リードポートC (診断用) | 35 | R | SHUT0 | PSTBM | SHUT1 | MCKEN | BUZ | TXRE | TXEE | RXRE | PORT Cの状態を読み取る。 |
| リードポートB | 33 | R | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | IMCK | EMCK | CDAT | PORT Bの状態を読み取る。 |
| リードポートA | 31 | R | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | SW3 | SW2 | SW1 | [illegible] |

(1) ポートの入出力信号

① ポートA入力

ディップスイッチSW2の状態をCPUのIN命令によって[illegible]

以下のデータは、PC-98XLのノーマルモード時にのみに使用される。

| データビット | 信号名 | 備考 |
|---|---|---|
| $D^7$ | [illegible] | [illegible] |
| $D^6$ | [illegible] | 未使用 |
| $D^5$ | SW6 | 未使用 |
| $D^4$ | SW5 | ON：メモリスイッチ保持　OFF：メモリスイッチ初期化 |
| $D^3$ | SW4 | ON：25行／画面　OFF：20行／画面 |
| $D^2$ | SW3 | ON：80文字／行　OFF：40文字／行 |
| $D^1$ | SW2 | ON：ターミナルモード　OFF：BASICモード |
| $D^0$ | [illegible] | [illegible] OFF |

# 第6章
# キーボード

## 6.1 キーボードインターフェイス

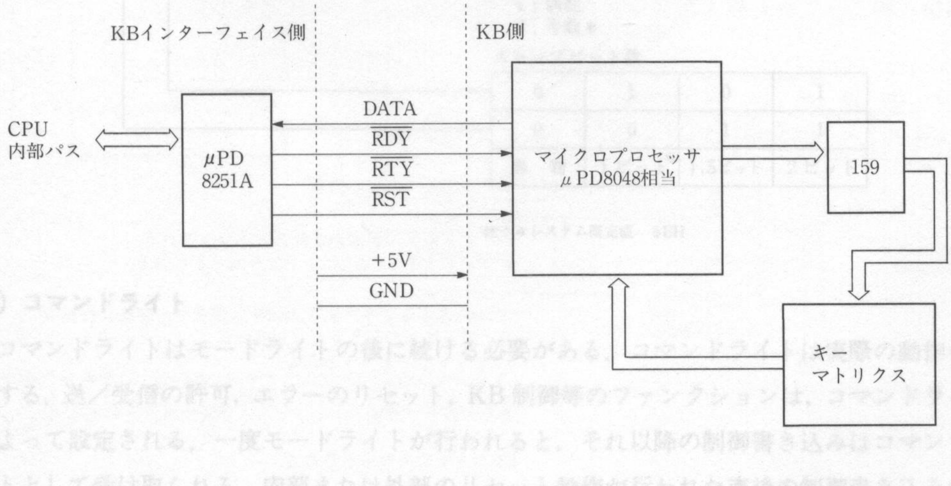
キーボード(以下KBと表記)はシステムユニットとは分離した別のユニットで構成されている. KBはシステムユニットの前面にあるコネクタとカールケーブルで接続されている. ケーブルにはKBとシステムユニット間の制御信号, データ信号の他にKBの+5V電源が含まれている. KBにはμPD8048マイクロプロセッサが内蔵されている. KBの107個のキーは大別すると, コントロールキー, プログラム可能なファンクションキー, シフトキー, ANKキーの4つのグループに分けられる.

システムユニット内のKBインターフェイス制御部には, KBからのビットシリアルなデータを受信してバイトパラレルなデータに変換するためにμPD8251Aが使用されている.

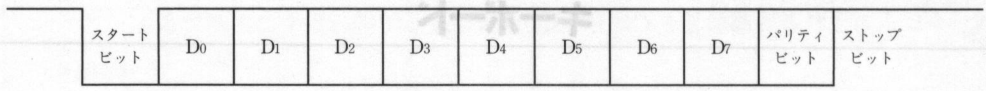
通信は調歩式(アシンクロナスモード)で行われ, シリアルデータにはキーコードとキー状態(キーが押下されている/キーが離されている)が直接読み出される. 転送速度は19.2KBPSである.

PC-98XLのノーマルモードにおいては, f・11 f・12 f・13 f・14 f・15 MOME キーを押してもBIOSで無視される.

### (1) KBインターフェイス ブロック図

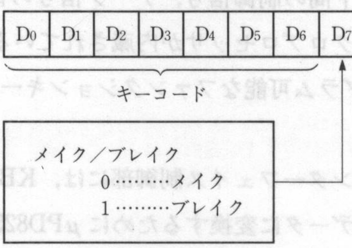


### (2) シリアルデータのビットフォーマット




- スタートビット ——— 1ビット
- データ長 ——— 8ビット
- パリティビット ——— ODDパリティ
- ストップビット ——— 1ビット

### (3) データフォーマット



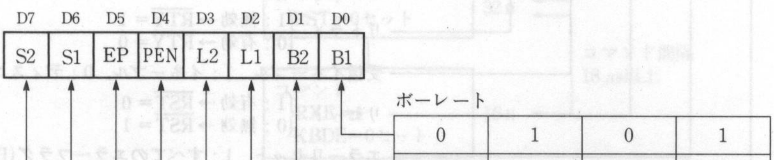

注：メイクはキーが押下された時の割り込みを示し，ブレイクは
キーが離された時の割り込みを示す．

## 6.2 I/Oアドレスと命令

| 命　　令 | I/Oポート アドレス | R/W | 機　　能 |
|---|---|---|---|
| (モード / コマンド) ライト | 43 | W | モードセット<br>コマンドセット |
| データ　リード | 41 | R | μPD8251Aにロードされた1バイトデータを読み出す |
| ステータス　リード | 43 | R | μPD8251Aのステータスを読み出す |

### (1) モードライト

μPD8251Aの一般的特性を設定するための動作である．μPD8251Aの内部または外部のリセット動作後に続く必要がある．モードライト後はコマンドライトが可能になる．




注：*システム既定値　5EH

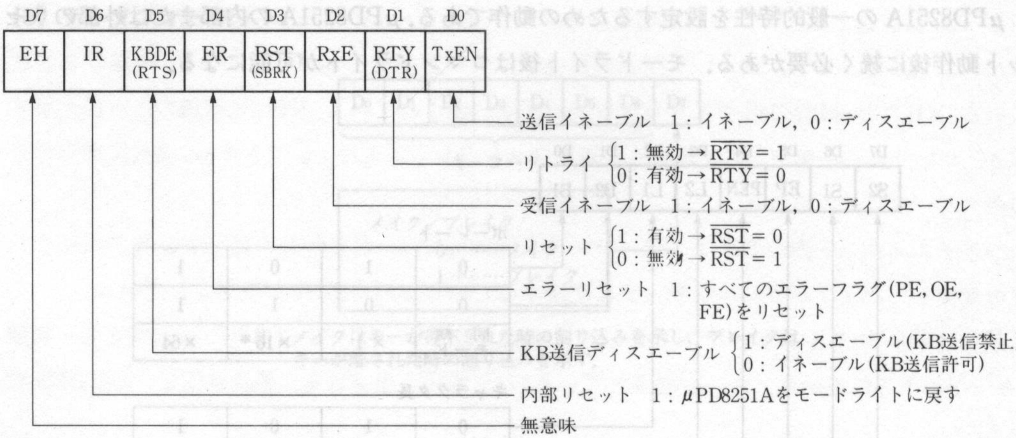
### (2) コマンドライト

コマンドライトはモードライトの後に続ける必要がある．コマンドライトは実際の動作を制御する．送／受信の許可，エラーのリセット，KB制御等のファンクションは，コマンドライトによって設定される．一度モードライトが行われると，それ以降の制御書き込みはコマンドライトとして受け取られる．内部または外部のリセット動作が行われた直後の制御書き込みのみがモードライトとなる．

$\mu$PD8251A の汎用出力ポートである $\overline{RTS}$，$\overline{DTR}$ を用いて，KB 送信ディスエーブル機能，KB インターフェイスの $\overline{RTY}$ 信号を実現している．また，$\mu$PD8251A から BREAK キャラクタを送出することにより，KB インターフェイスの $\overline{RST}$ 信号を作っている．したがって，

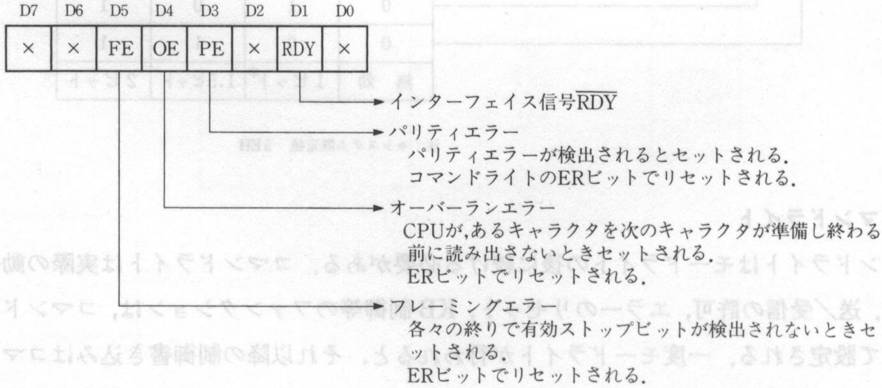
① KB 送信ディスエーブルのセット，リセット
② $\overline{RTY}$ 出力の ON，OFF
③ $\overline{RST}$ 出力 ON，OFF

を行う場合はコマンドライトを行わなければならない．

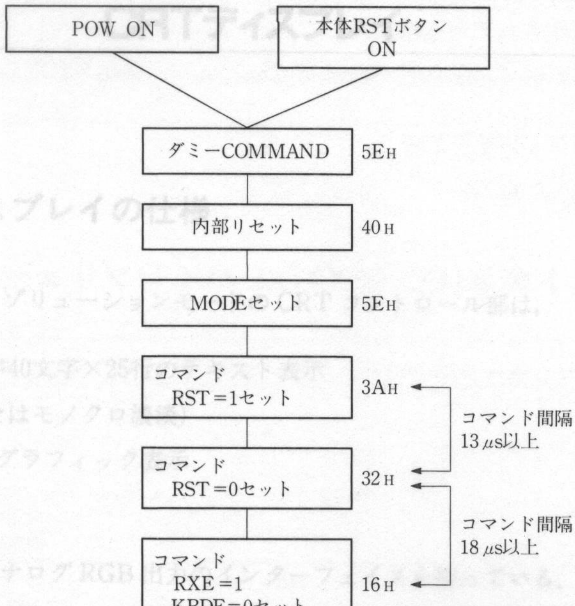


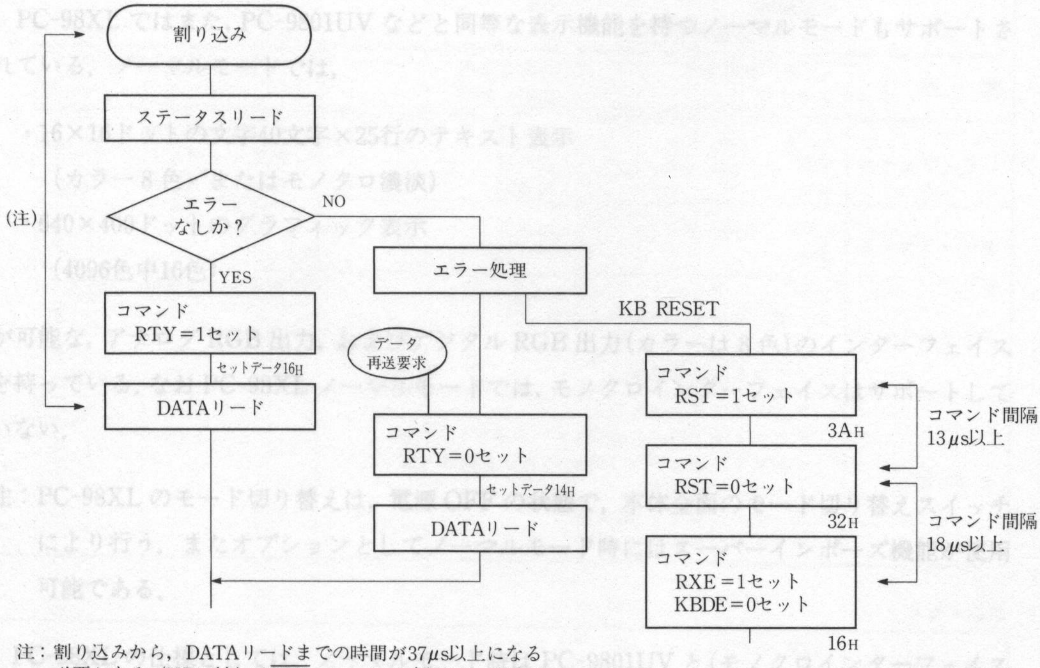

### (3) ステータスリード

$\mu$PD8251A は，動作中ならばいつでも，デバイスのステータスを読み出せる．ステータスの読み出し中はステータスの更新は禁止される．ステータスの更新は，ステータスに影響を与える事象が起こってから最大28クロック周期の遅延があることに注意すること．



## (4) 使用方法



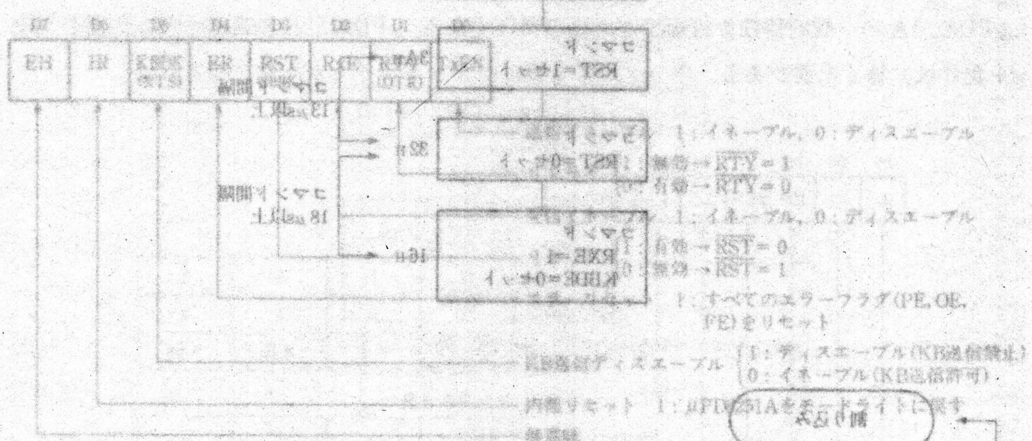




注：割り込みから，DATAリードまでの時間が37μs以上になる必要がある．(KBに対して$\overline{\text{RDY}}$＝1のパルス幅が37μs以上必要となる)

μPD8251A の汎用出力ポートである $\overline{RTS}$, $\overline{DTR}$ を用いて，KB 送信ディスエーブル機能と KB インターフェイスの $\overline{RTY}$ 信号を実現している．また，μPD8251A から BREAK キャラクタを送出することにより，KB インターフェイスの $\overline{RST}$ 信号を作っている．したがって，

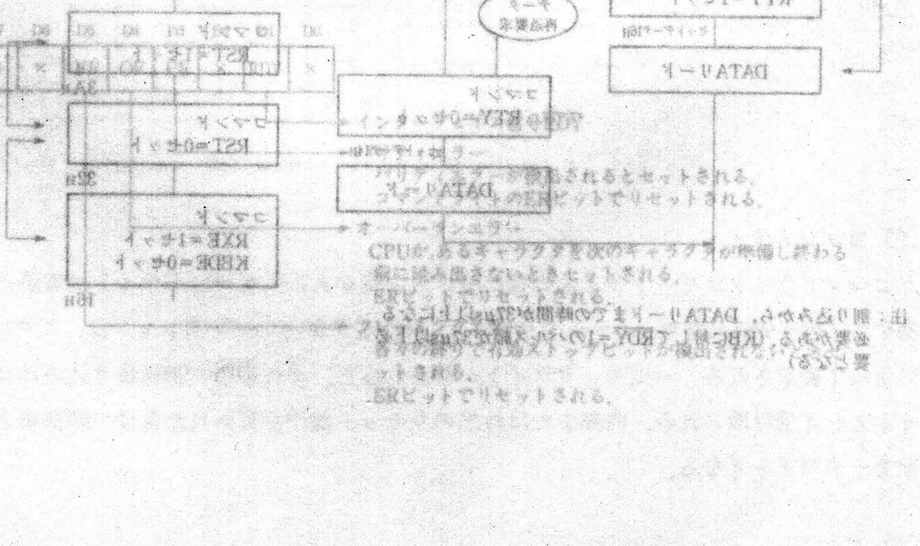
① KB 送信ディスエーブルのセット，リセット
② $\overline{RTY}$ 出力の ON，OFF
③ $\overline{RST}$ 出力 ON，OFF

を行う場合はコマンドライトを行わなければならない．




(3) ステータスリード

μPD8251A は，動作中ならばいつでも，デバイスのステータスを読み出せる．ステータスの読み出し中はステータスの更新は禁止される．ステータスの更新は，ステータスに影響を与える事象が起こってから最大28クロック周期の遅延があることに注意すること．



# 第7章

# CRTディスプレイ

## 7.1 CRT ディスプレイの仕様

PC-98XA/XL ハイレゾリューションモードのCRT コントロール部は,

・24×24ドットの文字40文字×25行のテキスト表示
　(カラー8色／またはモノクロ濃淡)
・1120×750ドットのグラフィック表示
　(4096色中16色)

が可能な,モノクロ／アナログ RGB 出力のインターフェイスを持っている.なお,この時の画面1ドットのパルス幅は20.9ns(47.846MHz)であるため,従来のKD-551や家庭用テレビ等は使用できず,表示には,N5924,N5923K,N5921 等が必要である.

PC-98XL ではまた,PC-9801UV などと同等な表示機能を持つノーマルモードもサポートされている.ノーマルモードでは,

・16×16ドットの文字40文字×25行のテキスト表示
　(カラー8色／またはモノクロ濃淡)
・640×400ドットのグラフィック表示
　(4096色中16色)

が可能な,アナログ RGB 出力,およびデジタル RGB 出力(カラーは8色)のインターフェイスを持っている.なお PC-98XL ノーマルモードでは,モノクロインターフェイスはサポートしていない.

注：PC-98XL のモード切り替えは,電源 OFF の状態で,本体全面のモード切り替えスイッチにより行う.またオプションとしてノーマルモード時にはスーパーインポーズ機能が使用可能である.

PC-98XL の仕様としては,ノーマルモード時は PC-9801UV と(モノクロインターフェイスを除く),ハイレゾリューションモード時は PC-98XA とほぼ同様であるが,次の点が異なっている.

**・ノーマルモード時**

CG ウィンドウを使用した文字フォントの Read/Write が可能

(PC-98XA で提供されている機能)

GRCG の強化(EGC : Enhanced Graphic Charger——強化された GRCG を採用)

**・ハイレゾリューションモード時**

GRCG の強化(EGC 採用)

## 7.2　テキスト表示(PC-98XA/XL ハイレゾリューションモード時)

PC-98XL ノーマルモード時の仕様は「PC-9800 シリーズテクニカルデータブック」を参照のこと.

### (1) 仕様

**表示画面**

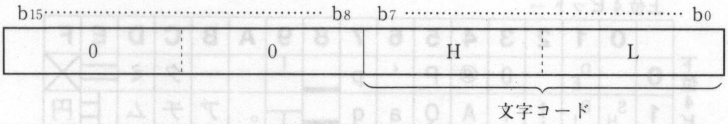
| | | | |
|---|---|---|---|
| **文　　字** | ANK 文字，特殊文字<br>半角文字(9, 10, 11区)<br>特殊文字(8, 12, 13, 14, 15区)<br>第Ⅰ水準漢字，非漢字<br>第Ⅱ水準漢字<br>拡張漢字<br>ユーザー定義文字(全角) | : 126+118<br>: 282(スペース含)<br>: 158<br>: 2,965+453(スペース含)<br>: 3,384<br>: 388<br>: 188 | ・ケイ線文字は，上下方向はつながらない<br>・1/4 角文字なし<br>・PC-98XAmodel 1，2，3 において拡張漢字はオプション |
| **表　　示** | ANK・半角文字<br>全角文字 | : 80桁×25行<br>: 40桁×25行 | 左記以外の設定不可 |
| **レターフェイス** | ANK<br>半角<br>全角 | : 10×18<br>: 10×18<br>: 24×24 | 全角文字表示時<br>桁間：4ドット<br>行間：6ドット<br>の隙間がある |
| **ボデーフェイス** | ANK・半角<br>全角 | : 14×30<br>: 28×30 | |
| **アトリビュート** | リバース，ブリンク，シークレット<br>アンダーライン，バーチカルライン<br>カラー8色(R・G・B)<br>モノクロ 8 階調の濃淡 | | キャラクタ単位に指定可 |
| **そ の 他** | ・スムーススクロールなし<br>・テキスト画面外の色指定(ボーダカラー)不可<br>・テキスト画面とグラフ画面のドット数は一致<br>・文字コードは JIS<br>・ユーザー定義文字188文字<br>・ライトペン使用不可 | | |

**画面メモリ**

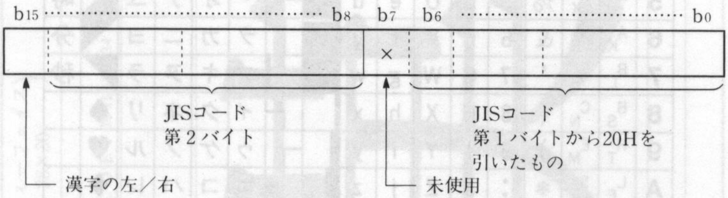
| | |
|---|---|
| **容　量** | 12K バイト(コード 8K，アトリビュート 4K)<br>アトリビュートの内 8 バイトは，バッテリーバックアップされる |
| **アクセス方式** | CPU による直接 R/W<br>GDC による描画機能は使用できない |
| **パリティ** | な　し |

### (2) 文字コード表現

①1バイト系 ANK

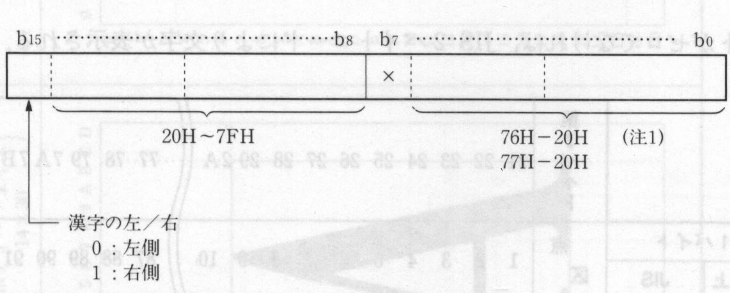



②2バイト系文字




漢字の VRAM 上の表現は4バイトで行われる.

③ユーザー定義文字




注1：76Hから20Hを引いた値

### (3) 表示文字種

①第2バイト($b_{15}$～$b_8$)がすべてゼロの場合，次の表の文字が表示される．

上位4ビット→　下位4ビット↓

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | DE | | 0 | @ | P | ‘ | p | | | | ー | タ | ミ | | ╳ |
| 1 | SH | D1 | ! | 1 | A | Q | a | q | | | 。 | ア | チ | ム | | 円 |
| 2 | SX | D2 | " | 2 | B | R | b | r | | | 「 | イ | ツ | メ | | 年 |
| 3 | EX | D3 | # | 3 | C | S | c | s | | | 」 | ウ | テ | モ | | 月 |
| 4 | ET | D4 | $ | 4 | D | T | d | t | | | 、 | エ | ト | ヤ | | 日 |
| 5 | EQ | NK | % | 5 | E | U | e | u | | | ・ | オ | ナ | ユ | | 時 |
| 6 | AK | SN | & | 6 | F | V | f | v | | | ヲ | カ | ニ | ヨ | | 分 |
| 7 | BL | EB | ' | 7 | G | W | g | w | | | ァ | キ | ヌ | ラ | | 秒 |
| 8 | BS | CN | ( | 8 | H | X | h | x | | | ィ | ク | ネ | リ | ♠ | |
| 9 | HT | EM | ) | 9 | I | Y | i | y | | | ゥ | ケ | ノ | ル | ♥ | |
| A | LF | SB | * | : | J | Z | j | z | | | ェ | コ | ハ | レ | ♦ | |
| B | HM | EC | + | ; | K | [ | k | { | | | ォ | サ | ヒ | ロ | ♣ | |
| C | CL | → | , | < | L | ¥ | l | ¦ | | | ャ | シ | フ | ワ | ● | ＼ |
| D | CR | ← | - | = | M | ] | m | } | | | ュ | ス | ヘ | ン | ○ | |
| E | SO | ↑ | . | > | N | ^ | n | ~ | | | ョ | セ | ホ | ゛ | | |
| F | SI | ↓ | / | ? | O | _ | o | | | | ッ | ソ | マ | ゜ | | |

②第2バイトがゼロでなければ，JIS 2バイトコードにより文字が表示される．

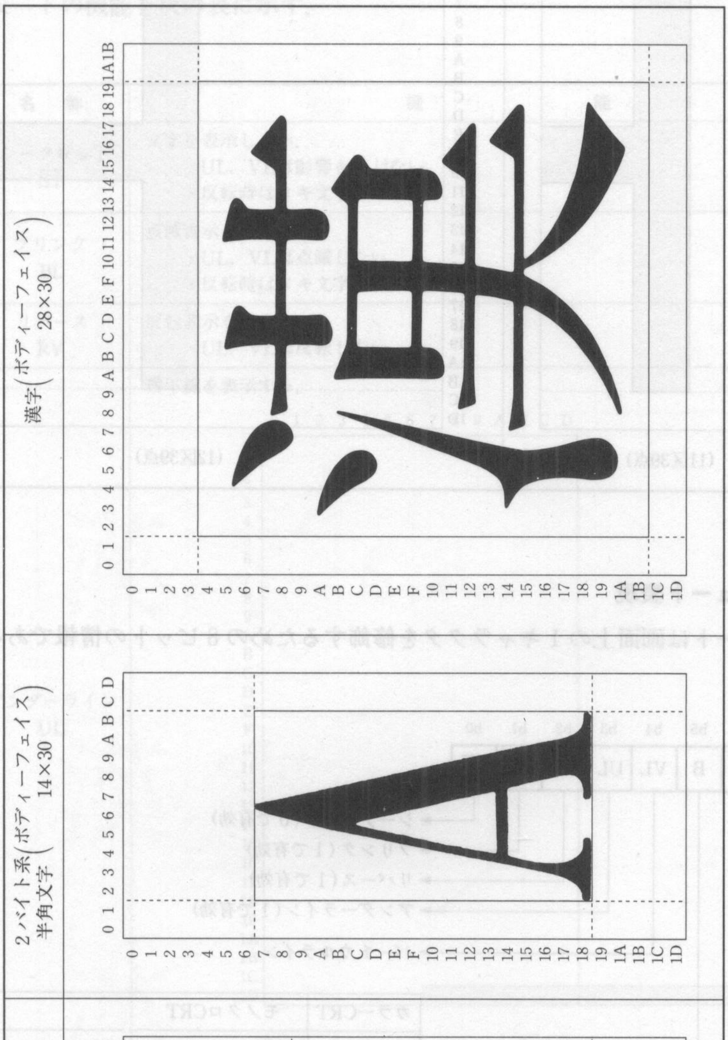
| 第2バイト | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 77 | 78 | 79 | 7A | 7B | 7C | 7D | 7E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 点 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 |

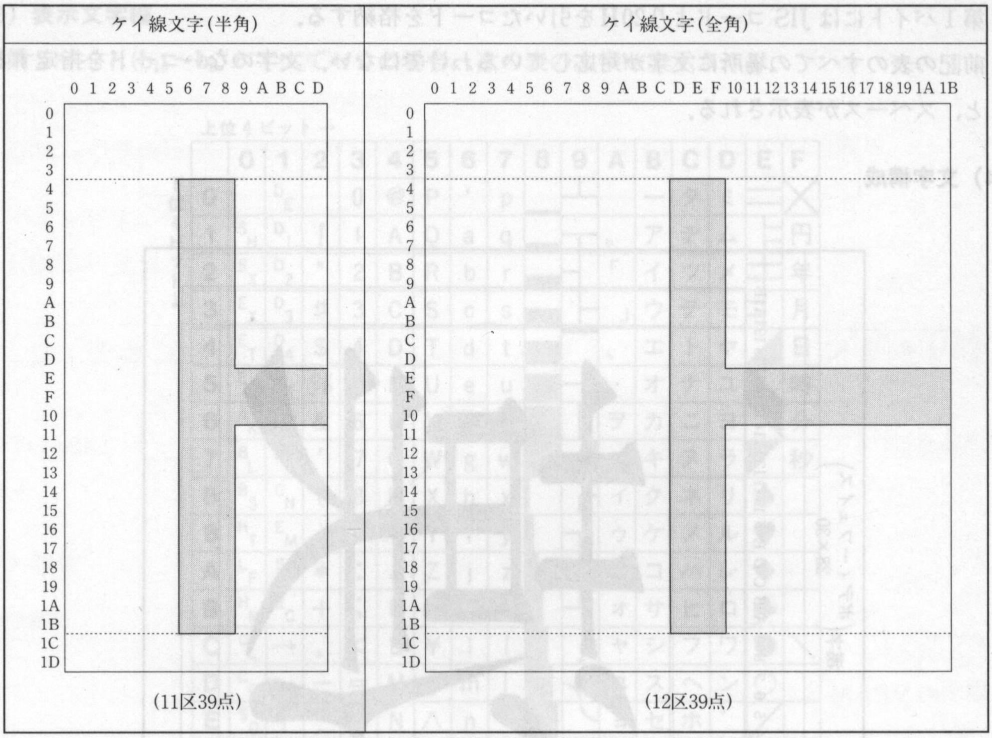
| 第1バイト メモリ上 | 第1バイト JIS | 区 | |
|---|---|---|---|
| 01 | 21 | 1 | |
| 02 | 22 | 2 | |
| 03 | 23 | 3 | |
| 04 | 24 | 4 | |
| 05 | 25 | 5 | |
| 06 | 26 | 6 | |
| 07 | 27 | 7 | |
| 08 | 28 | 8 | |
| 09 | 29 | 9 | この部分は半角表示(14×30)となる．その他はすべて全角． |
| 0A | 2A | 10 | |
| 0B | 2B | 11 | |
| 0C | 2C | 12 | |
| 0D | 2D | 13 | |
| 0E | 2E | 14 | |
| | 4B | 44 | |
| 2C | 4C | 45 | |
| 2E | 4E | 46 | |
| 2F | 4F | 47 | |

・第1バイトにはJISコードより20Hを引いたコードを格納する．

・前記の表のすべての場所に文字が対応しているわけではない．文字のないコードを指定すると，スペースが表示される．

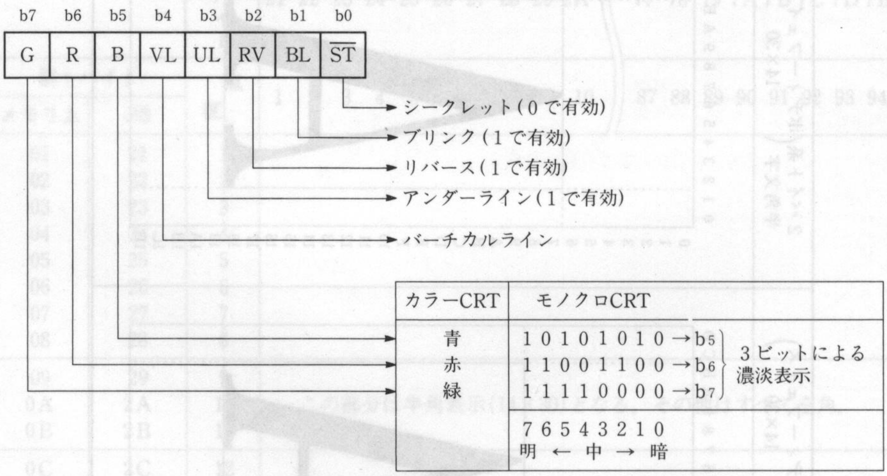
（4） 文字構成





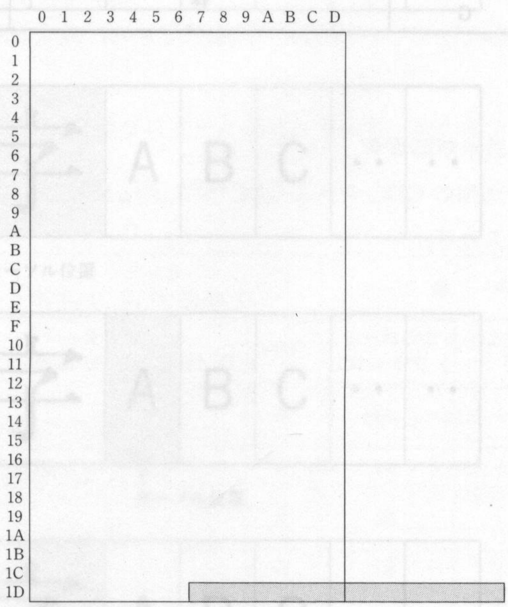

## (5) アトリビュート表現

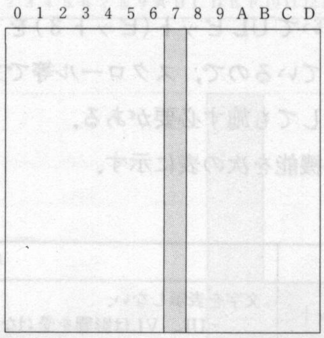
アトリビュートは画面上の1キャラクタを修飾するための8ビットの情報である.



アトリビュートは画面上のすべての文字に対して8ビットがいつでも有効であるため，その文字の属性を書込む必要がある．たとえば，アンダーラインを継続して表示する場合には，それらの文字全部についてULビット(ビット3)を1にしなければならない．またアトリビュートは各文字に対応しているので，スクロール等で文字を移動させた場合は，それと同じ移動をアトリビュートに対しても施す必要がある．

アトリビュートの機能を次の表に示す．

| ビット位置 | 名　称 | 機　　　能 |
|---|---|---|
| 0 | シークレット<br>$\overline{\mathrm{ST}}$ | 文字を表示しない．<br>・UL，VLは影響を受けない．<br>・反転時はヌキ文字が消える． |
| 1 | ブリンク<br>BL | 点滅表示を行う．<br>・UL，VLは点滅しない．<br>・反転時はヌキ文字が点滅する． |
| 2 | リバース<br>RV | 反転表示を行う．<br>・UL，VLは反転しない． |
| 3 | アンダーライン<br>UL | 横下線を表示する．<br><br>ULは必ず半カラム右にずれる．ただし，80カラム目の右半分はカットされる．<br>色指定により，その色が出る． |

| ビット位置 | 名　称 | 機　能 | |
|---|---|---|---|
| 4 | バーチカル<br>ライン<br>VL | 縦線を表示する。<br><br> | |
| | | カラーCRTの色指定 | モノクロCRTの濃度指定 |
| 5 | ブルー<br>B | 青 | $2^0$ |
| 6 | レッド<br>R | 赤 | $2^1$ |
| 7 | グリーン<br>G | 緑 | $2^2$ |

### (6) カーソル表示

①形状

カーソルは基本的にはブリンキングブロック形式であるが，GDC に対するプログラムによって自由な形に設定することができる．シークレット領域でもカーソルは表示されて，その領域の色指定に従う．

ただしインタレース制御が乱れるため，カーソルブリンキングレートの設定を行ってはならない．

②漢字表示時のカーソル表示

全角文字の後半をカーソル位置にすると，後半の部分の1カラムのみにカーソル表示される．したがって漢字の時のカーソル移動は2カラムずつとする．



### (7) テキスト VRAM の使用法

テキスト VRAM は CPU，GDC に対して次に示すようなメモリ空間を持っている．

| GDC アドレス | CPU アドレス | HIGH<br>$b_{15}$ ―― $b_8$ | LOW<br>$b_7$ ―― $b_0$ |
|---|---|---|---|
| 0000 | E 0000 | テキスト<br>文字<br>1ページ | テキスト<br>文字<br>1ページ |
| 0800 | E 1000 | テキスト<br>文字<br>2ページ | テキスト<br>文字<br>2ページ |
| 1000 | E 2000 | ╳ | アトリビュート<br>1ページ |
| 1800 | E 3000 | ╳ | アトリビュート<br>2ページ |

↑ワードアドレス　↑バイトアドレス　↑メモリは存在しない

GDC とのインターフェイス（表示開始アドレス，カーソル表示位置）は GDC アドレスで行う．

### (8) テキスト表示制御命令

テキスト制御用の GDC である GDC（マスタ）μPD7220 及び各制御回路は，次表に示す命令によって制御される．

| 命　　令 | I/O ポートアドレス | R/W | データ<br>$D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | 備　　　考 |
|---|---|---|---|---|
| リードステータス | 60 | R | GDC<br>←ステータスフラグ→ | |
| ライトパラメータ | 60 | W | ← GDC パラメータ → | |
| リードデータ | 62 | R | ← GDC データ →<br>（ライトペン） | |
| ライトコマンド | 62 | W | ← GDC コマンド → | （注1） |
| CRT インタラプトリセット | 64 | W | × × × × × × × × | （注2） |
| ライトモードレジスタ | 68 | W | 0 0 0 0 ／ Mode F/F ←→：ADR2 ADR1 ADR0 DT | 次図参照 |

注1：GDC（M）は描画機能がないので，描画に関するコマンド（描画制御コマンド）は使用出来ない．
注2：CRT の垂直同期ごとの割り込みをリセットする．CRTV はパワーオン後1回割り込むが，リセットを出さない限り，その後は割り込まない．

Mode F/F

| Mode F/F | NAME | A D R 2 | A D R 1 | A D R 0 | D T 1 | D T 0 | 主として関係する部分(注1) |
|---|---|---|---|---|---|---|---|
| 5 | KAC Mode | 1 | 0 | 1 | 漢字アクセスモード | | K |
| | | | | | ビットマップ | コードアクセス | |
| 6 | NVMW PERMIT | 1 | 1 | 0 | 不揮発メモリへの書き込み | | T |
| | | | | | PERMIT | INHIBIT | |
| 7 | DISP(注2) ENABLE | 1 | 1 | 1 | 表示可とする | すべての画面を表示しない | T |

注1：T：テキスト，K：漢字
注2：このF/Fはリセット SW 押下時に画面を消去する働きをする．リセット SW が押されると，GDC の CLK 等が供給されなくなるので，イニシャライズルーチンが立上がるまで画面は不確定となる．GDC のイニシャライズが終ったあと，本F/Fを〝1〞とすることによって，表示可能な状態となる．

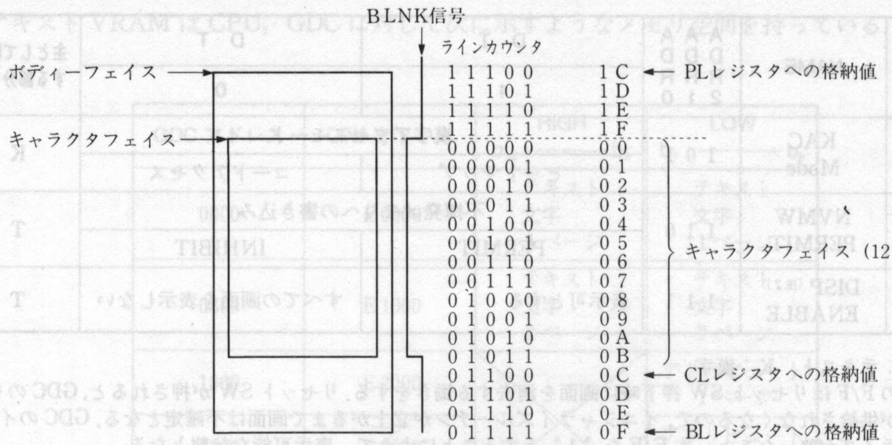
### (9) ラインカウンタ制御命令

ラインカウンタ制御回路は，CG ラインカウンタ出力，アンダーラインのタイミング出力，スムーススクロール機能を実現するためのアドレス加算回路へのタイミング出力等，CRT の垂直方向の制御信号を出力する．

ラインカウタ制御回路はライトだけが可能なレジスタを持っていて，次の表に示す命令によって制御する．

| 命　令 | I/O ポートアドレス | R/W | データ $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | 備　　考 |
|---|---|---|---|---|
| ライト　PL | 70 | W | キャラクタ位置ライン数 (PL)(注1) | ラインカウンタの初期値（ボディーフェイスのうち，キャラクタが表示される位置の上から数えたライン数の2の補数） |
| ライト　BL | 72 | W | ボディーフェイスライン数 (BL)(注1) | キャラクタの先頭を0とした時のボディーフェイス下端のライン数 |
| ライト　CL | 74 | W | キャラクタライン数 (CL)(注1) | キャラクタフェイスのライン数 |
| ライト SSL | 76 | W | スムーススクロールライン数 (SSL)(注2) | スクロールエリア内の文字がスクロールしているライン数 |
| ライト SUR | 78 | W | スクロールエリア上辺位置行数 (SUR)(注2) | スクロールエリアの上辺の位置の行数の2の補数（この次の行よりスクロールする） |
| ライト SDR | 7A | W | スクロールエリア行数 (SDR)(注2) | （スクロールエリアの行数）－1 |

注1：ラインカウンタ制御回路設定例




注2：ハードウェアでは，スムーススクロールをサポートしていないため，
　　　これらのレジスタを不用意にアクセスしてはならない．

### (10) CGウィンドウ

文字(ANKおよび漢字)フォントのREAD/WRITEはメモリ空間にマッピングされたCGウィンドウにより行う.




ウィンドウに割当てる文字は次のI/Oポートに出力する.

| 命　　令 | I/Oアドレス | R/W |
|---|---|---|
| WRITE 2nd byte code | 0A1H | W |
| WRITE 1st byte code | 0A3H | W |

**注意**

1．前記コードはテキストVRAMに格納するコードと同一である．

2．ユーザー定義文字にフォントを登録する場合も1．の原則に従い，前記ポートに7620H-2000Hから765FH-2000Hをセットした後，メモリライトを行う．

3．CGのアクセスは次のライトオペレーションにより2つのモードがある．

| 命　　令 | I/O アドレス | R/W |
|---|---|---|
| WRITE　mode F/F | 68H | W |

| WRITE　mode F/F | DATA<br>$b_7$ $b_6$ $b_5$ $b_4$ $b_3$ $b_2$ $b_1$ $b_0$ |
|---|---|
| ビットマップモード | 0 0 0 0 1 0 1 1 |
| コードアクセスモード | 0 0 0 0 1 0 1 0 |

| モード | テキスト表示 | CG ウィンドウアクセス |
|---|---|---|
| ビットマップ | 無 | 随　　時 |
| コードアクセス | 有 | GDC VSNC 中のみ |

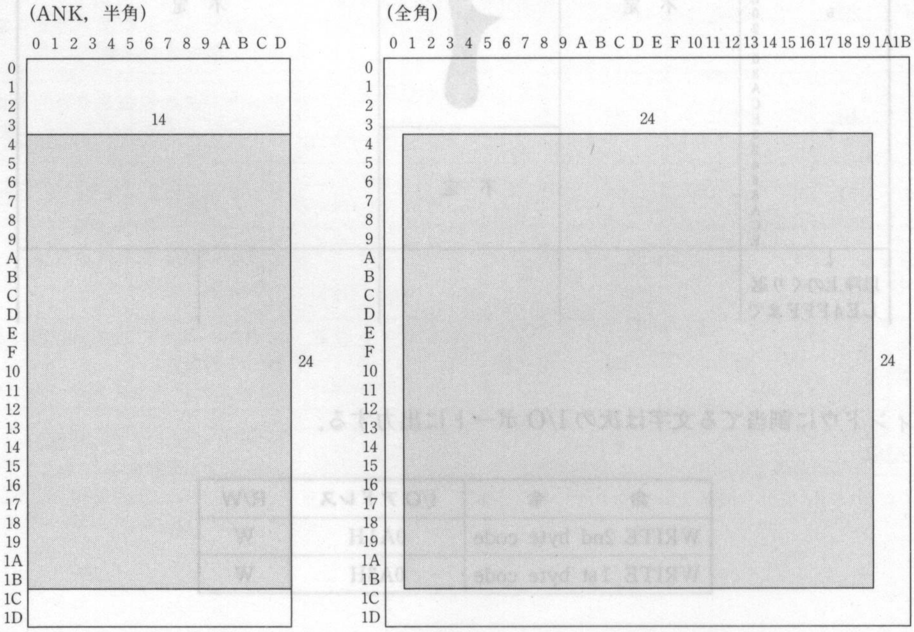
ただし，WRITE 1st byte code, 2nd byte codeはモードにかかわらず随時行える．VSNC=1は12.5ms中，152$\mu$sである．

4．CGウィンドウのメモリアクセスは2 waitである．

5．コードアクセスモードでフォントの読み取りがVSNC中で終らない時は，一時的にビットマップモードにし，表示開始までにコードアクセスモードに戻すのはかまわない．表示開始はVSNCの立上がりから395$\mu$sである．

6．CGウィンドウにより，アクセスされるデータ．

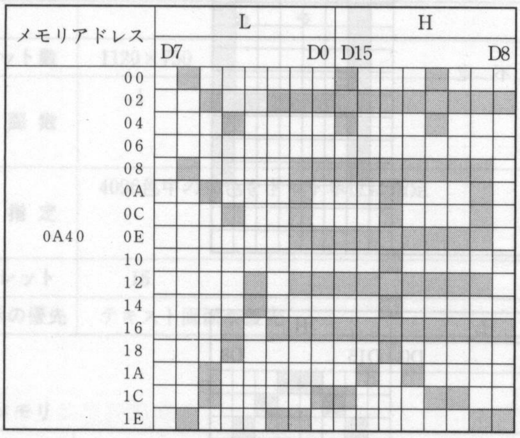
(ANK，半角)　　(全角)



### (11) PC-98XLノーマルモード時のCGウィンドウ

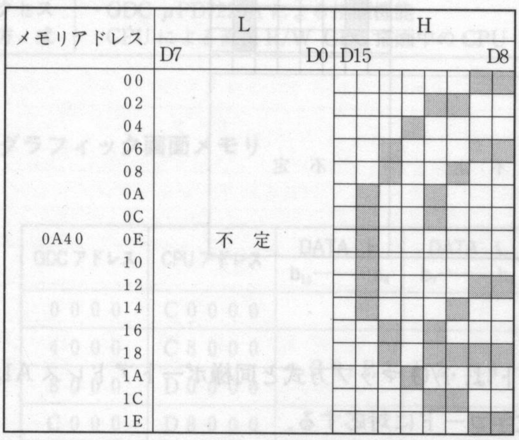
PC-98XLのノーマルモード時は，CGウィンドウを通して，文字フォントのREAD/WRITEを実行することが出来る．

全角文字




第１バイトが以下のものを除く
・2C～2F：全角ケイ線特殊文字
・76：ユーザー定義
・79～7C：拡張漢字

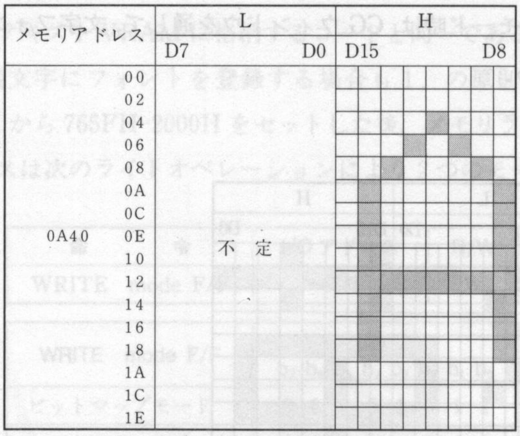
全角文字(半角アクセス)



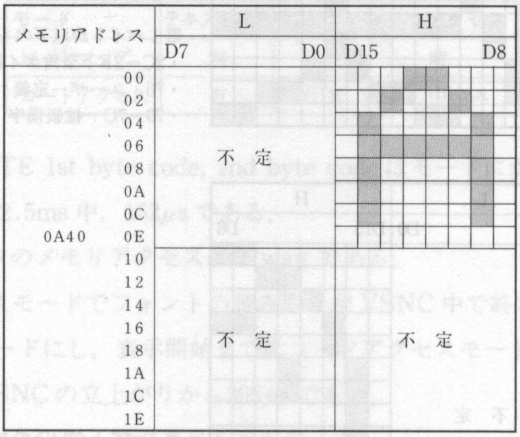

第１バイトが以下のもののみ
・2C～2F：全角ケイ線特殊文字
・76：ユーザー定義
・79～7C：拡張漢字

文字の左右はポートアドレスA5に対する出力データのL／Rビットにより指定される．

半角文字，88モードANK文字




80モード ANK文字




**注意**

1．READ/WRITE される文字フォントは，I/O マップ方式と同様ポートアドレス A1, A3 に対する OUT 命令で設定された文字コードに対応する．

2．CG ウィンドウの READ/WRITE は，I/O マップ方式と同様．ポートアドレス 68H のビットマップ／コードアクセスモードによる．コードアクセスモード時，READ/WRITE は，VSNC 中のみ可能．

ただし，ANK CG の読みだしは，モードに関わらず VSNC 中のみ可能．

3．CG ウィンドウは 0A4000H から 0A4FFFH までの空間に割り当てられているが，0A420H 以上のアドレスはおなじパターンの繰り返しとなる．

4．CG ウィンドウに対して DMA 転送してはならない．

# 7.3 グラフィック表示

## (1) 仕様

**表示画面**

| | カ ラ ー | モ ノ ク ロ |
|---|---|---|
| **ドット数** | 1120×750 | 同 左 |
| **画 面 数** | 1 | 1<br>(パレットの設定方法によって，4画面として使用できる) |
| **色 指 定** | 4096色中の16色をドット単位に指定 | 16階調濃淡(パレットからのR・G・B出力信号(各4ビット)の最上位とGのみ$2^2$ビットとの値による)をドット単位に指定 |
| **パレット** | 16 | 同 左 |
| **表示の優先** | テキスト画面が優先 | 同 左 |

**画面メモリ**

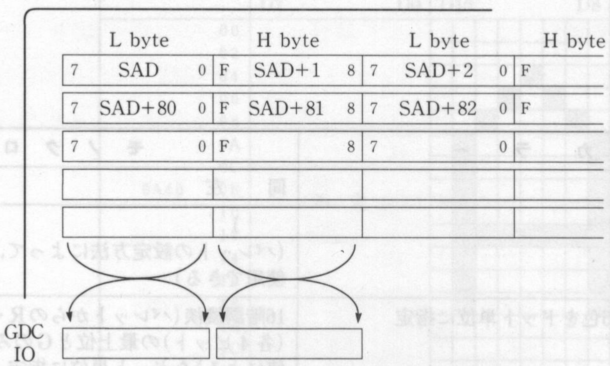
| | |
|---|---|
| **容 量** | 128Kバイト×4プレーン |
| **アクセス方 式** | ・GDC μPD7220A による描画機能<br>・CPU による直接 R/W GDC 描画中の CPU R/W は不可 |

## (2) グラフィック画面メモリ

| GDCアドレス | CPUアドレス | DATA H $b_{15}$——$b_8$ | DATA L $b_7$——$b_0$ |
|---|---|---|---|
| 0000 | C0000 | | |
| 4000 | C8000 | P0, P1, P2, P3 | |
| 8000 | D0000 | | |
| C000 | D8000 | | |

→ バイトアドレス(アドレスの最下位により H byte, L byte を切分ける)
→ ワードアドレス(H byte, L byte を必ず同時にアクセス)

メモリ内データは次のようにCRTに表示される.




GDC経由でデータをR/Wする時はFirst DotがLSBになる.

### (3) グラフィック制御命令

グラフィック制御用のGDCであるGDC(スレーブ)$\mu$PD7220および各制御回路は，次表に示す命令によって制御される. なお，アドレスA4H, A6Hは，グラフィックチャージャ(GRCG)のレジスタである.

| 命　　令 | I/Oポート アドレス | R/W | データ $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|
| リード　ステータス | A0 | R | GDCステータスフラグ | | | | | | | |
| ライト　パラメータ | A0 | W | GDCパラメータ | | | | | | | |
| リード　データ | A2 | W | GDCデータ | | | | | | | |
| ライト　コマンド | A2 | R | GDCコマンド | | | | | | | |
| ライト　モードレジスタ (注1) | A4 | W | CG | RMW | RP | RP | P3 EN | P2 EN | P1 EN | P0 EN |
| ライト　タイルレジスタ | A6 | W | (注2) | | | | | | | |
| ライト　パレットアドレス | A8 | W | 0 | 0 | 0 | 0 | P3 | P2 | P1 | P0 |
| ライト　パレットGREEN | AA | W | 0 | 0 | 0 | 0 | G3 | G2 | G1 | G0 |
| ライト　パレットRED | AC | W | 0 | 0 | 0 | 0 | R3 | R2 | R1 | R0 |
| ライト　パレットBLUE | AE | W | 0 | 0 | 0 | 0 | B3 | B2 | B1 | B0 |

注1：モードレジスタ

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| CG mode | RMW mode | RP1 | RP0 | $\overline{P3}$ EN | $\overline{P2}$ EN | $\overline{P1}$ EN | $\overline{P0}$ EN |

b3〜b0：
- CG mode＝0の時，Write時のプレーン選択[注4](ビット対応)
- CG mode＝1の時，GRCGのアクティブプレーン

b5〜b4：
- CG mode＝0の時，Read時のプレーン選択

| RP1 | RP0 | |
|---|---|---|
| 0 | 0 | P0 read |
| 0 | 1 | P1 read |
| 1 | 0 | P2 read |
| 1 | 1 | P3 read |

- CG mode＝1の時，"00"をSetする

b6：
- CG mode＝0の時，"0"をSetする
- CG mode＝1の時，0：TDW，TCR mode
  1：RMW mode

b7：
- 0：normal mode
- 1：CG mode[注3]

注2：タイルレジスタ

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| tile pattern 0－3 | | | | | | | |

mode Regにwriteするとtile Reg 0 writeにリセットされ，その後書き込む度にtile Reg 1，tile Reg 2，tile Reg 3，tile Reg 0…となる．

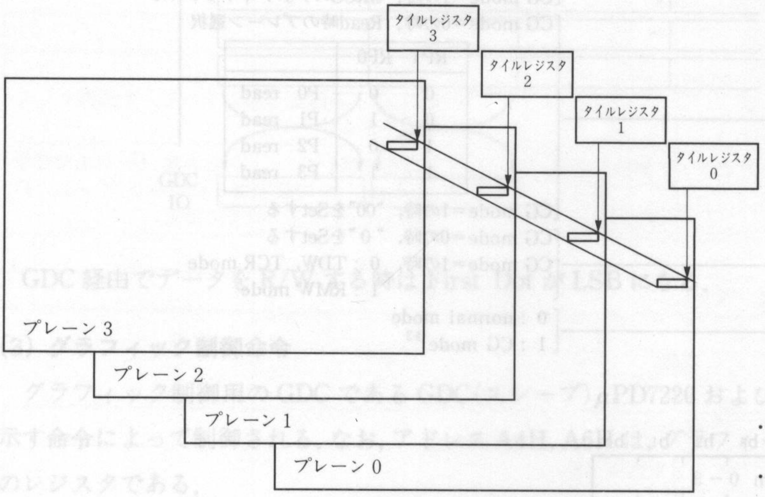
注3：グラフィックチャージャ動作中(CG mode)はGDCからの描画は禁止される．
GDCに描画命令を出しても，VRAMへの直接アクセスは起こらない．
モードレジスタはグラフィックチャージャ動作中は変更できない

注4：この設定はCGmodo＝0の時はGDCの描画に対しても有効
(READプレーンとWRITEプレーンを変えて設定すると，プレーン間の転送が可)

### (4) GRCGのモード

a)TDWモード

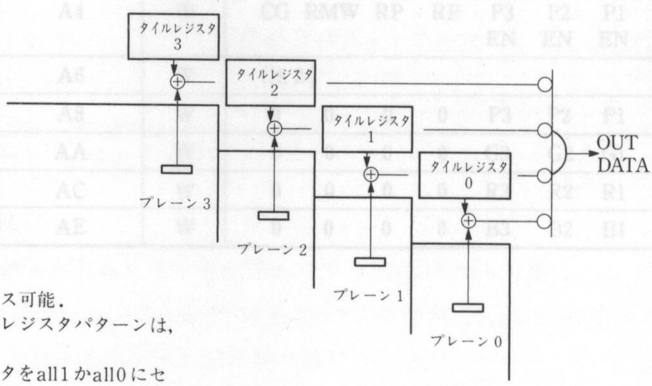
VRAMに対してCPUがライトすると，CPUのライトデータは無視され，タイルレジスタの内容が各プレーンにライトされる．




・STRING命令，バイトアクセス可能．
・ワードアクセス時，タイルレジスタパターンはL, Hに拡張される．

b)TCRモード

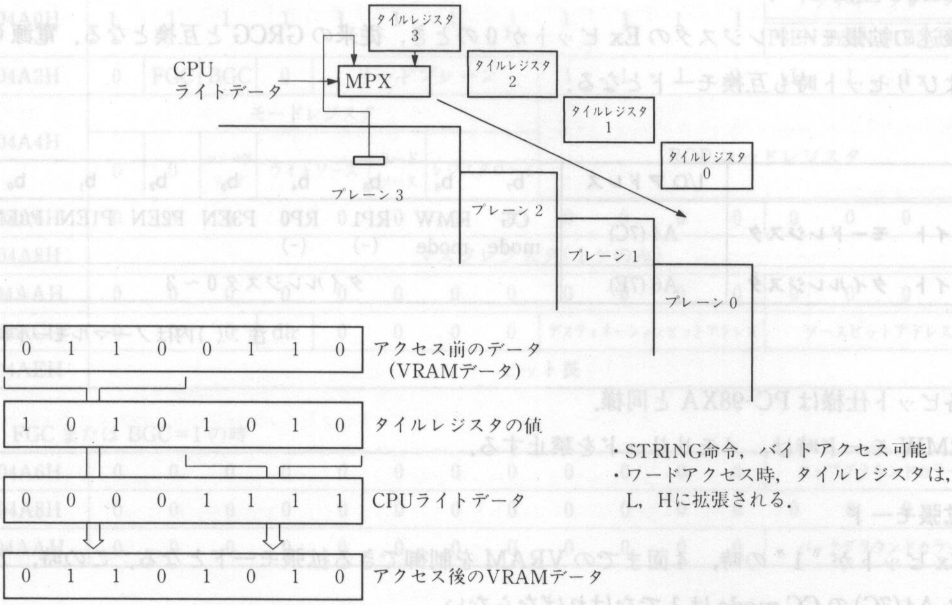
CGモード設定後，該当アドレスをCPUがリードすると，各プレーンとタイルレジスタの一致をとり，すべてのプレーンで一致のとれたビットを1にしてCPUにリードデータとして出力する．




・SCAN命令，バイトアクセス可能．
・ワードアクセス時，タイルレジスタパターンは，H，Lに拡張される．
・色の検出ではタイルレジスタをall1かall0にセットする．
・アクティブにしないプレーンは無視される．

c)RMWモード

RMWモード設定後,CPUがVRAMにライトすると,ライトデータが1のビットはタイルレジスタの内容がライトされ,0のビットは元のデータが残される.




### (5) EGC(Enhanced Graphic Charger)

PC-98XLのグラフィックチャージャには,PC-98XAのGRCGに次の機能が追加された.

- PC-98XA(PC-9801UV, VF, VM)のGRCGとの互換モード有り(ただし,GDCからも使用可能).
- 4面のVRAMをハード的に同時制御可能.
- ラスターオペレーション可能.
- VRAM内のデータ移動のため,ビット単位のシフト機能を持つ.

EGCは,CPUとVRAMの間に介在し,CPUが読み込み・書き出しを行う際のデータを捉え,データの加工を行うものである.EGCを機能させるには,EGCのレジスタをセットした後,CPUに〝MOVS〟などの命令を実行させる.するとCPUが読み込み動作を行った時にVRAMから送られてくるデータをEGCが取り込み,次にCPUが書き出し動作を行う際に,EGCは書き出しの対象となったVRAMからデータの読み出しを行い,以前からVRAMに有ったデータとCPUが書き込むデータ(およびEGC内部のレジスタ内のデータ)との間でラスタオペレーション(ビットパターン間での演算:以後ROPと記述)を行い,その結果をVRAMに書き出す.

実際の動作では，動作の対象となるプレーンを指定する，指定された色に対応するビットだけを読み出す，ROPの対象となるデータを切り変える，EGC内部のレジスタをシフトさせる，などの指定が可能であり，組み合せによって様々な動作が可能である．

①GRCG互換モード

後述の拡張モードレジスタのExビットが0のとき，従来のGRCGと互換となる．電源ONおよびリセット時も互換モードとなる．

| | I/Oアドレス | $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|
| ライト　モードレジスタ | A4 (7C) | CG mode | RMW mode | RP1 (-) | RP0 (-) | P3EN | P2EN | P1EN | P0EN |
| ライト　タイルレジスタ | A6 (7E) | タイルレジスタ0～3 | | | | | | | |

注：( )内はノーマルモード時．

各ビット仕様はPC-98XAと同様．

RMWモード時は，メモリリードを禁止する．

②拡張モード

Exビットが〝1〞の時，4面までのVRAMを制御できる拡張モードとなる．この時，アドレスA4(7C)のCG modeは1でなければならない．

| I/Oポート | 76543210 (Exビット) | モード |
|---|---|---|
| 6A | 00000101 | 拡張モード |
| | 00000100 | 互換モード |

Exビットは，mode F/F(アドレス6AH)においてReg Write Enable中でなければ変更できない．またモード切り替えは，GDC描画中に行ってはならない．

以下にNormal，互換，拡張の各モードでのCGモードおよびExビットの状態を示す．

| モード | CGモード | Exビット |
|---|---|---|
| Normalモード | 0 | 0 |
| 互換モード | 1 | 0 |
| 拡張モード | 1 | 1 |

注：Exビット＝0にするときは，必ずその前に04A0HのEGCレジスタに0FFF0Hをセットしなければならない．

拡張モード時のレジスタを以下に示す．

| I/Oアドレス | データ $D_{15}$ | $D_{14}$ | $D_{13}$ | $D_{12}$ | $D_{11}$ | $D_{10}$ | $D_9$ | $D_8$ | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 04A0H | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | アクティブプレーン<br>P3EN | P2EN | P1EN | P0EN |
| 04A2H | 0 | FGC | BGC | 0 | リードプレーン | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 04A4H | モードレジスタ<br>0 | 0 | コンペアリード | ライトソース | | リードソース | レジスタロード | | ROPコードレジスタ | | | | | | | |
| 04A6H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 04A8H | マスクレジスタ（1ケのみ） | | | | | | | | | | | | | | | |
| 04AAH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 04ACH | 0 | 0 | 0 | dir | 0 | 0 | 0 | 0 | デスティネーションビットアドレス | | | | ソースビットアドレス | | | |
| 04AEH | ビット長 | | | | | | | | | | | | | | | |

FGCまたはBGC=1の時

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 04A6H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | フォアグラウンドカラー | | | |
| 04A8H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 04AAH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | バックグラウンドカラー | | | |

・アクティブプレーン

命令が適用されるプレーンをビット対応で設定する．すべての命令に優先する．

ライト系 ──〝1〞のプレーンは変更されない．

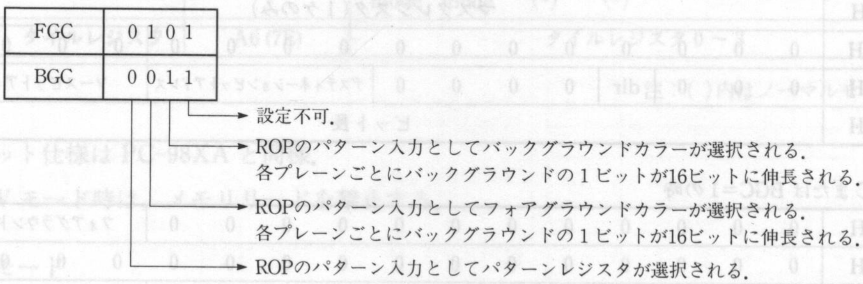
リード系 ── コンペアリードビット＝1の時に有効．

〝1〞のプレーンはコンペアされない．

・リードプレーン

コンペアリードビット＝0の時に読み込むプレーンを設定する．

・FGC，BGC




・マスクレジスタ

全プレーンに対しレジスタ1つ．

0のビットはデータ書き込み後も変更されない．

ROPに優先される．

・フォアグラウンドカラー

FGC指定時のフォアグラウンドカラーを設定する．

・バックグラウンドカラー

BGC指定時のバックグラウンドカラーを設定する．

・モードレジスタ

| 0 | 0 | コンペア<br>リード | ライト<br>ソース | リード<br>ソース | レジスタ<br>ロード |
|---|---|---|---|---|---|

レジスタロード：
- 00：パターンレジスタは変更なし．
- 01：メモリリード命令時にデータをロード<br>(GDCアクセス時無効)
- 10：メモリライト時に，ライトされるメモリの変更前のデータをロード
- 11：設定不可

リードソース：
- 0：シフタの入力はVRAMデータ<br>CPUのリードでデータを取り込み，次のライトでROPを行う
- 1：シフタの入力はCPUデータ<br>CPUのライト時に，そのCPUデータがROPされる．

ライトソース：
- 00：CPUライトデータ
- 01：ROPの演算結果
- 10：パターンレジスタ
- 11：設定不可

コンペアリード：
- 0：コンペアリードを行わない．
- 1：コンペアリードを行う．<br>コンペアはフォアグラウンドカラーと行う．

・ROPコードレジスタ

ROPの演算内容をビットパターンにより設定する．

ビットパターン：R7 R6 R5 R4 R3 R2 R1 R0　R：オペレーション結果

S：S入力(MOVSのSource) D：D入力(MOVSのDestination) P：P入力(パターンレジスタ)とすると，

$$R=R7\cdot PSD+R6\cdot \overline{P}SD+R5\cdot PS\overline{D}+R4\cdot \overline{P}S\overline{D}+R3\cdot P\overline{S}D+R2\cdot \overline{PS}D+R1\cdot P\overline{SD}+R0\cdot \overline{PSD}$$

(例) $R=D\oplus S$

$$=\overline{D}\cdot S+D\cdot \overline{S}$$

$$=\overline{D}\cdot S\ (P+\overline{P})\ +D\overline{S}\ (P+\overline{P})$$

$$=PS\overline{D}+\overline{P}S\overline{D}+P\overline{S}D+\overline{PS}D$$

↓

R7＝0，R6＝0，R5＝1，R4＝1，R3＝1，R2＝1，R1＝0，R0＝0

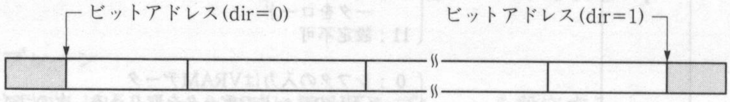
ROPコード＝(00111100)

・ソースビットアドレス，デスティネーションビットアドレス(GDCアクセス時は無効)

VRAMのデータを転送する場合のワード内のビットアドレスを設定する.

dir=0の時，ワードの先頭から数える.

dir=1の時，ワードの最後から数える.




・ビット長

転送するビット長を設定する.

1ラスタ方向のみ．ただし，内容は保存される.

・Dir

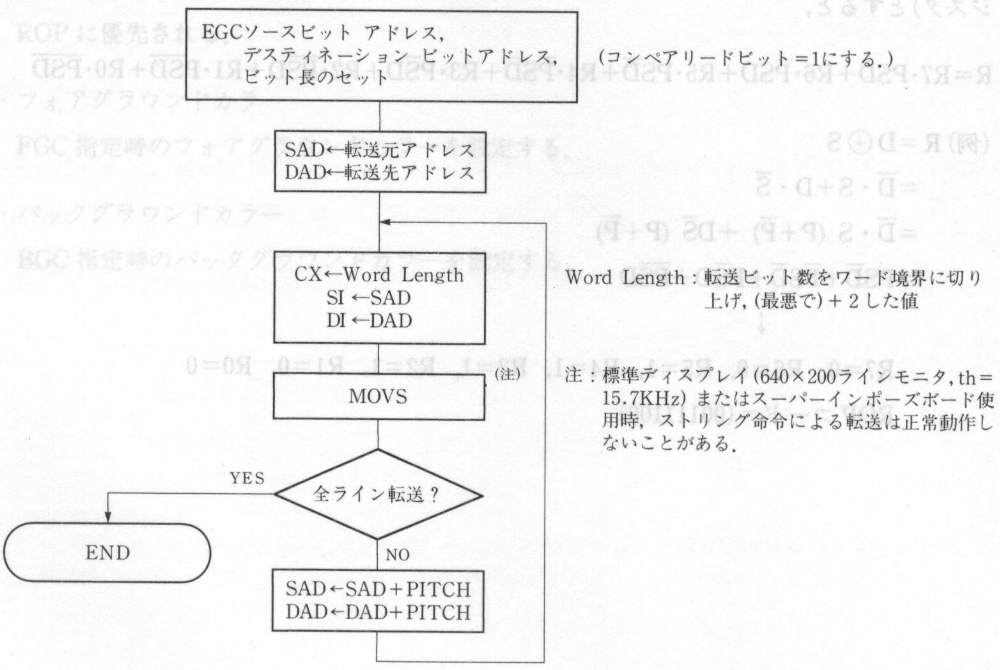
転送方向の設定

dir=0：アドレスの低い方から転送

dir=1：アドレスの高い方から転送

EGCの動作において，dirの設定はビットboundaryの先頭と最後のMASK処理に影響する.

・転送のさせ方



・転送時のビットアドレスとビット長の設定

bit-length　：転送したいビット数

word-length：転送したいエリアを含んだワード数

S-BITAD　：ソースビットアドレス

D-BITAD　：デスティネーションビットアドレス

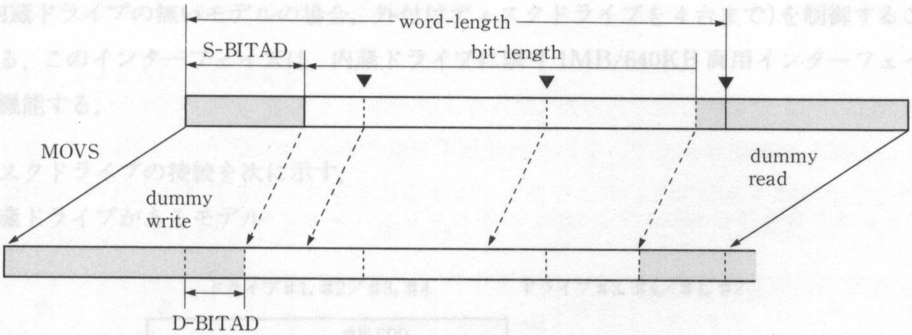
ただし，dir＝0（MOVS の SI，DI を増加させる）とした場合.

a）S-BITAD＞D-BITAD の時

転送先の開始番地は，目的の場所－1とする.

さらに，（S-BITAD＋bit-length）／16＝（D-BITAD＋bit-length）／16の時

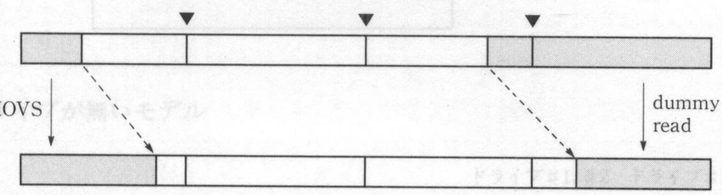
word-length を＋1する.



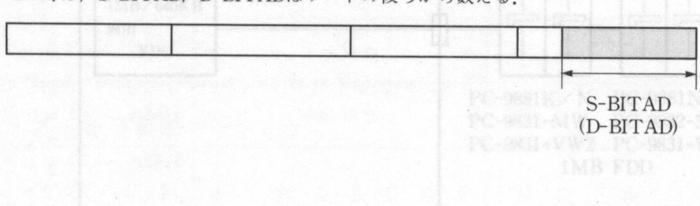

b）S-BITAD＜D-BITAD の時

（S-BITAD＋bit-length）／16＜（D-BITAD＋bit-length）／16の時

word-length を＋1する.




dir＝1の時は，S-BITAD，D-BITADはワードの後ろから数える.

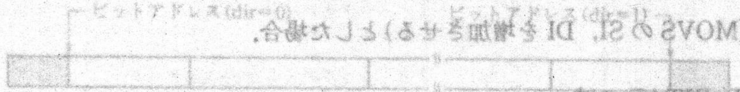


・ソースビットアドレス，デスティネーションビットアドレスの設定

VRAMのデータを転送する場合のワード内のビットアドレスを設定する．

dir＝0の時，ワードの先頭から数える．

dir＝1の時，ワードの最後から数える．




・ビット長

転送するビット長を設定する．

1ラスタ方向のみ．ただし，内容は保存される．

・Dir

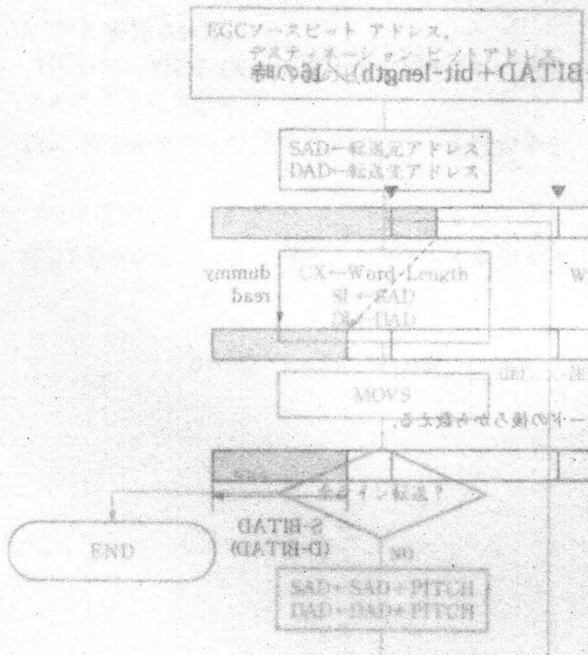
転送方向の設定

dir＝0：アドレスの低い方から転送

dir＝1：アドレスの高い方から転送

EGCの動作において，dirの設定はビットboundaryの先頭と最後のMASK処理に影響する．

・転送のさせ方




Word Length：転送ビット数をワード単位に切り上げ，(最速で)＋2した値

注：標準ディスプレイ(640×200ライン モニタ，fh＝15.7KHz)またはスーパーインポーズボード使用時にはEGCは動作しないことがある．

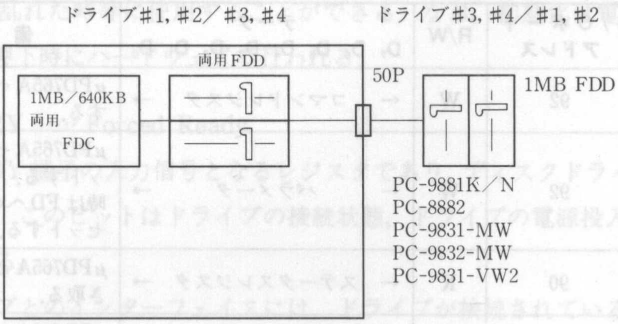
# 第8章
# ディスクインターフェイス

## 8.1 1MB フロッピーディスク

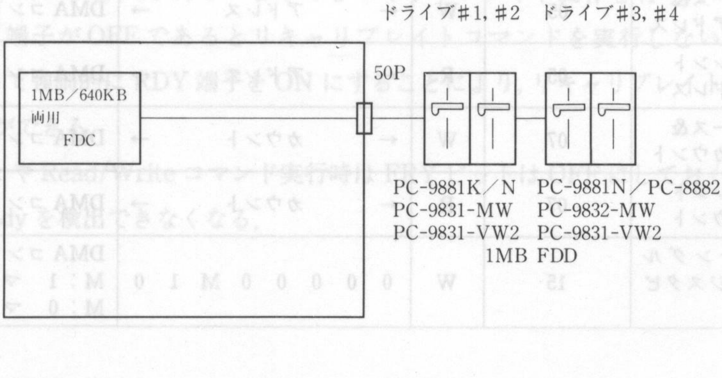
PC-98XL/XA は，1MB/640KB 両用フロッピーディスクを制御するためのインターフェイスを内蔵している．このインターフェイスは制御用 LSI として μPD765A を使用しており，内蔵する両用フロッピーディスクドライブ 2 台と外付けの 1MB フロッピーディスクドライブ 2 台(内蔵ドライブの無いモデルの場合，外付けディスクドライブを 4 台まで)を制御することが出来る．このインターフェイスは，内蔵ドライブに限り 1MB/640KB 両用インターフェイスとして機能する．

ディスクドライブの接続を次に示す．

・内蔵ドライブがあるモデル




・内蔵ドライブが無いモデル



データ転送はDMAにより行われる．DMAの制御はDMAコントローラに対しコマンドを発行することにより行う．本インターフェイスで使用するDMAと割り込みは次のとおりである．

| | PC-98XL ノーマルモード | | PC-98XA, PC-98XL ハイレゾリューションモード |
|---|---|---|---|
| | 1MB | 640KB | |
| DMA チャネル | 2 | 3 | 1 |
| 割り込みレベル | IR11 | IR10 | IR11 |
| I/O ポート | 1M FDC 専用 | 640KB FDC 専用 | 共用 |

注：PC-98XL ノーマルモード時は、PC-9801UV/VM と同様．
　　ハイレゾリューションモード時は PC-98XA と同様．

なお，PC-98XAの場合DMAチャネルおよび割り込みレベルは，専用スロットにのみ接続されている．そのためこのインターフェイスを他のスロットに接続することはできない．また，PC-9801UV/VMで使用可能であった1MBインターフェイスボード(PC9801-15)は，拡張スロットにDMA制御信号が出ていないため使用不可能である．

**I/O アドレスと命令(PC-98XA/PC-98XL ハイレゾリューションモード)**

| 命　令 | I/O ポートアドレス | R/W | データ $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | 備　考 |
|---|---|---|---|---|
| ライトコマンド | 92 | W | ← コマンドレジスタ → | μPD765A へコマンドをセットする． |
| ライトレジスタ | 92 | W | ← パラメータ → | μPD765A へのパラメータをセットする．NON-DMA モード時は FD への書き込みデータもセットする． |
| リードステータス | 90 | R | ← ステータスレジスタ → | μPD765A からステータスを引き取る． |
| リードデータ | 92 | R | ← リザルトステータス → | μPD765A からリザルトステータスを引き取る．NON-DMA モード時には FD から読み取ったデータも引き取る． |
| ライトベース&カレントアドレス | 05 | W | ← アドレス → | DMA コントローラ |
| リードカレントアドレス | 05 | R | ← アドレス → | DMA コントローラ |
| ライトベース&カレントカウント | 07 | W | ← カウント → | DMA コントローラ |
| リードカレントワードカウント | 07 | R | ← カウント → | DMA コントローラ |
| ライトシングルマスクレジスタビット | 15 | W | 0 0 0 0 0 M 1 0 | DMA コントローラ<br>M：1　マスクオン<br>M：0　マスクオフ |

| 命令 | I/Oポートアドレス | R/W | データ $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | 備考 |
|---|---|---|---|---|
| ライトモードレジスタ | 17 | W | 0 1 0 0 $m_1$ $m_2$ 1 0 | DMAコントローラ<br>$m_1m_2$：00 ベリファイ転送<br>〃 ：01 メモリライト転送<br>〃 ：10 メモリリード転送 |
| クリアバイトポインタフリップフロップ | 19 | W | × × × × × × × × | DMAコントローラ |
| ライトDMAチャネル#1バンク | 21 | W | × × × × ←バンク→ | DMAコントローラ |
| ライトコントロールレジスタ | 94 | W | RST FRY HD × × × × × | μPD765Aの外部レジスタへのセット |
| リードコントロールレジスタ | 94 | R | MODE 0 HD 0 × × × × | |

## (1) ライトコントロールレジスタ

● D7ビット：RST …… Reset

μPD765AのRESET端子の入力信号となるレジスタであり，μPD765Aを初期化するのに使用する．初期化はμPD765Aへのコマンド，パラメータの転送シーケンスやリザルトステータス転送シーケンスが乱れた時等に使用することができる．なお，初期化は電源投入直後およびRESETスイッチ押下時にハードウェアで行われる．

● D6ビット：FRY …… Forced Ready

μPD765AのRDY端子の入力信号となるレジスタであり，ディスクドライブのRDY信号と論理和されている．このビットはドライブの接続状態，ドライブの電源投入状態をチェックするために使用される．

ディスクドライブとのインターフェイスには，ドライブが接続されているか否か，電源が投入されているか否かの状態を直接示す信号線は無い．そこでドライブにリキャリブレイト動作をさせてTrack00信号が返って来たら，そのドライブは接続かつ電源投入状態であると判定する(ドライブはディスクが挿入されていなくてもリキャリブレイト動作を行う)．この際，μPD765AはRDY端子がOFFであるとリキャリブレイトコマンドを実行しない．そこでFRYビットによって強制的にRDY端子をONにすることにより，リキャリブレイトコマンドを実行させることができる．

なお通常のSeekやRead/Writeコマンド実行時はFRYビットはOFFにしておくこと．さもないとNot Readyを検出できなくなる．

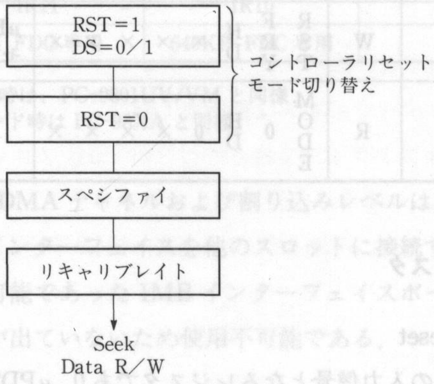
● D 5 ビット：HD…… High Density

1MB モードか 640KB モードかを指定する．

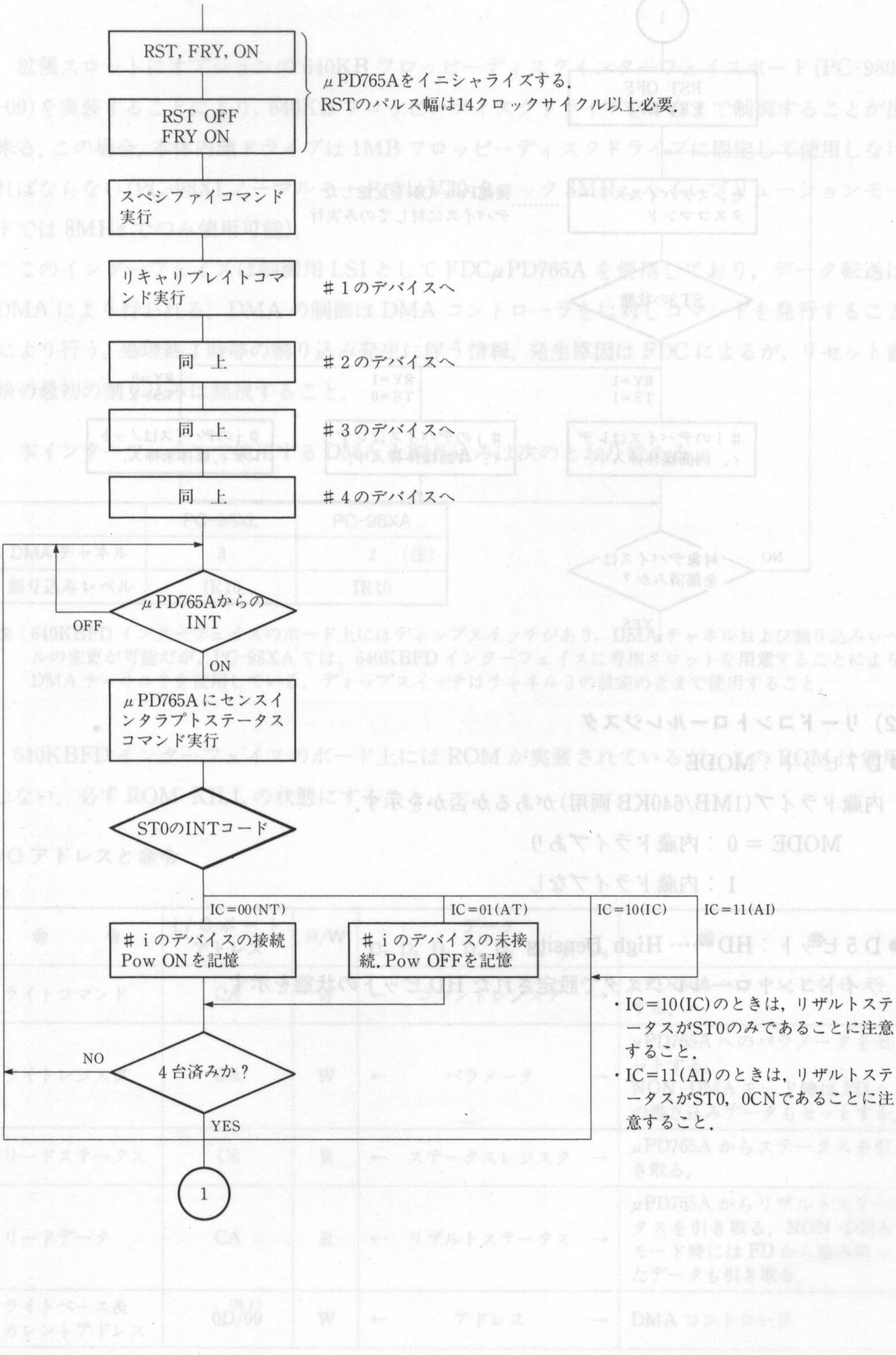
HD＝0：1MB モード

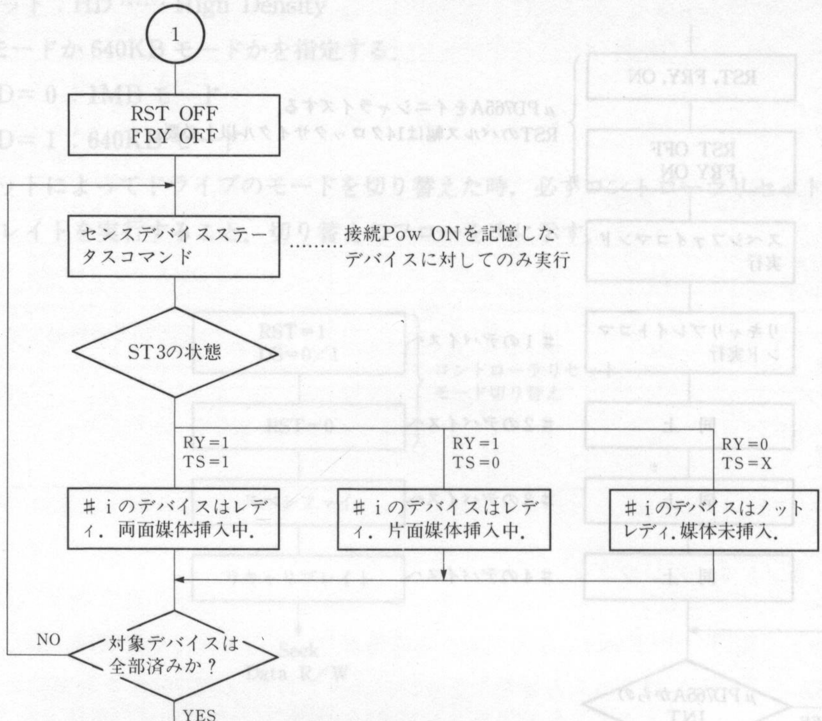
HD＝1：640KB モード

HD ビットによってドライブのモードを切り替えた時，必ずコントローラリセットおよびリキャリブレイトを実行すること．切り替えのフローを次に示す．




デバイス接続状態，デバイス電源 ON/OFF 状態，媒体マウント状態のチェック方法の例を次図に示す．


RST, FRY, ON
μPD765Aをイニシャライズする.
RSTのパルス幅は14クロックサイクル以上必要.
RST OFF
FRY ON
スペシファイコマンド実行
リキャリブレイトコマンド実行
＃1のデバイスへ
同　上
＃2のデバイスへ
同　上
＃3のデバイスへ
同　上
＃4のデバイスへ
μPD765AからのINT
OFF
ON
μPD765Aにセンスインタラプトステータスコマンド実行
ST0のINTコード
IC＝00(NT)
IC＝01(AT)
IC＝10(IC)
IC＝11(AI)
＃ｉのデバイスの接続 Pow ONを記憶
＃ｉのデバイスの未接続. Pow OFFを記憶
・IC＝10(IC)のときは，リザルトステータスがST0のみであることに注意すること.
・IC＝11(AI)のときは，リザルトステータスがST0，0CNであることに注意すること.
NO
4台済みか？
YES
1




### (2) リードコントロールレジスタ

● D 7ビット：MODE

内蔵ドライブ(1MB/640KB 両用)があるか否かを示す．

MODE ＝ 0 ：内蔵ドライブあり

1 ：内蔵ドライブなし

● D 5ビット：HD …… High Density

ライトコントロールレジスタで設定されたHDビットの状態を示す．

## 8.2 640KBフロッピーディスク

拡張スロットにオプションの640KBフロッピーディスクインターフェイスボード(PC-9801-09)を実装することにより，640KBフロッピーディスクドライブを4台まで制御することが出来る．この場合，本体内蔵ドライブは1MBフロッピーディスクドライブに固定して使用しなければならない(PC-98XLノーマルモードではV30クロック8MHz，ハイレゾリューションモードでは8MHzでのみ使用可能)．

このインターフェイスは制御用LSIとしてFDCμPD765Aを使用しており，データ転送はDMAにより行われる．DMAの制御はDMAコントローラをに対しコマンドを発行することにより行う．処理終了時等の割り込み発生に伴う情報，発生原因はFDCによるが，リセット直後の最初の割り込みは無視すること．

本インターフェイスで使用するDMAと割り込みは次のとおりである．

| | PC-98XL | PC-98XA |
|---|---|---|
| DMAチャネル | 3 | 2　(注) |
| 割り込みレベル | IR10 | IR10 |

注：640KBFDインターフェイスのボード上にはディップスイッチがあり，DMAチャネルおよび割り込みレベルの変更が可能だが，PC-98XAでは、640KBFDインターフェイスに専用スロットを用意することによりDMAチャネル2を使用している．ディップスイッチはチャネル3の設定のままで使用すること．

640KBFDインターフェイスのボード上にはROMが実装されているが，このROMは使用しない．必ずROM KILLの状態にすること．

### I/Oアドレスと命令

| 命　令 | I/Oポートアドレス | R/W | データ $D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ | 備　考 |
|---|---|---|---|---|
| ライトコマンド | CA | W | ← コマンドレジスタ → | μPD765Aへコマンドをセットする． |
| ライトレジスタ | CA | W | ← パラメータ → | μPD765Aへのパラメータをセットする．NON-DMAモード時はFDへの書き込みデータもセットする． |
| リードステータス | C8 | R | ← ステータスレジスタ → | μPD765Aからステータスを引き取る． |
| リードデータ | CA | R | ← リザルトステータス → | μPD765Aからリザルトステータスを引き取る．NON-DMAモード時にはFDから読み取ったデータも引き取る． |
| ライトベース&カレントアドレス | 0D/09 (注1) | W | ← アドレス → | DMAコントローラ |

| 命　　令 | I/Oポート<br>アドレス | R/W | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 備　　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| リードカレントアドレス | (注1)<br>0D/09 | R | ← | | | アドレス | | | | → | DMAコントローラ |
| ライトベース&<br>カレントカウント | (注1)<br>0F/0B | W | ← | | | カウント | | | | → | DMAコントローラ |
| リードカレント<br>ワードカウント | (注1)<br>0F/0B | R | ← | | | カウント | | | | → | DMAコントローラ |
| ライトシングル<br>マスクレジスタビット | 15 | W | 0 | 0 | 0 | 0 | 0 | M | 1 | 1 | DMAコントローラ<br>M：1　マスクオン<br>M：0　マスクオフ |
| ライトモード<br>レジスタ | 17 | W | 0 | 1 | 0 | 0 | $m_1$ | $m_2$ | 1 | 1 | DMA部のレジスタ<br>$m_1m_2$：00ベリファイ転送<br>〃　：01メモリライト転送<br>〃　：10メモリリード転送 |
| クリアバイトポインタ<br>フリップフロップ | 19 | W | × | × | × | × | × | × | × | × | DMAコントローラ |
| ライトDMAチャネルバンク | (注1)<br>25/23 | W | × | × | × | × | ← | バン | ク | → | DMAコントローラ |
| ライトコントロール | CC | W | RST | × | × | DMAE | MTON | TMSK | × | TTRG | μPD765Aの外部レジスタへのセット |
| リードスイッチ<br>シグナル | CC | R | FINT1 | FINT0 | DMACH | RDY | × | × | × | × | |

注1：DMAのアドレスは機種により異なる．左がXL，右がXAのものである．

### (1) ライトコントロールレジスタ

● D7ビット：RST …… Reset

μPD765AのRESET端子の入力信号となるレジスタであり，μPD765Aを初期化するのに使用する．初期化はμPD765Aへのコマンド，パラメータの転送シーケンスやリザルトステータス転送シーケンスが乱れた時等に使用することができる．なお，初期化は電源投入直後およびRESETスイッチ押下時にハードウェアで行われる．

● D4ビット：DMAE …… DMA Enable

DMAチャネル#2または#3を使ってデータ転送を行う時，DMAコントローラからのDRQ信号，DACK信号を許可するフリップフロップである．DMA使用時のみONにし，使用しない時はOFFとすること．

● D3ビット：MTON …… MOTOR ON

ディスクドライブのMOTOR ON端子の入力信号を保持する．

本FDCボードで制御されるすべてのディスクドライブのモータを同時にON/OFFするのに使用する．

● D2ビット：TMSK …… Timer Interrupt Mask

FDCからのタイマ割り込みをマスクするためのレジスタであり，本ビットを〝0〞にすることにより，FDCからのタイマ割り込みはマスクされる．

パワーON直後は〝0〞であり，タイマ未使用時は〝0〞にしておくこと．

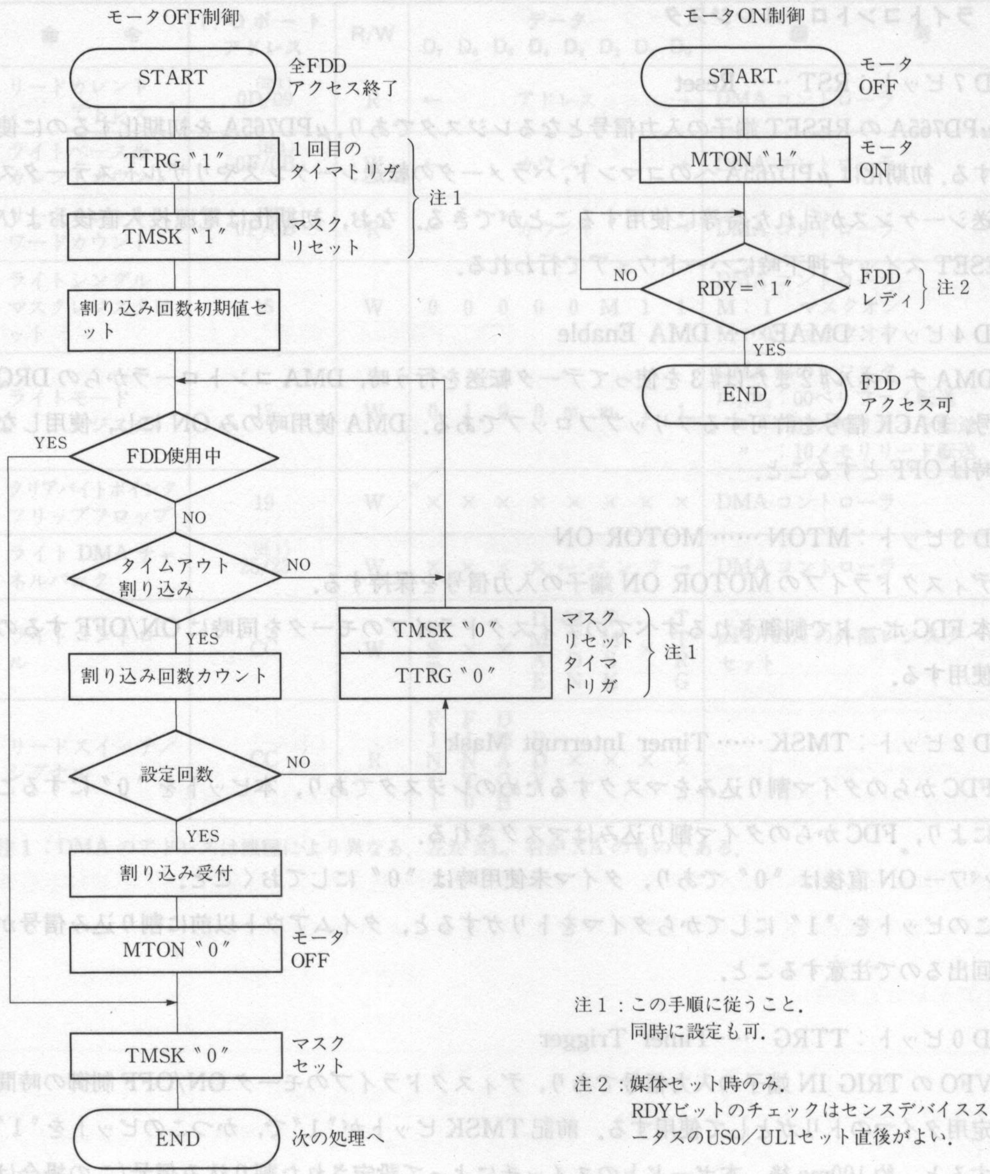
このビットを〝1〞にしてからタイマをトリガすると，タイムアウト以前に割り込み信号が1回出るので注意すること．

● D0ビット：TTRG …… Timer Trigger

VFOのTRIG IN端子の入力信号であり，ディスクドライブのモータON/OFF制御の時間設定用タイマのトリガとして使用する．前記TMSKビットが〝1〞で，かつこのビットを〝1〞にすると，約100ms後，本ボード上のスイッチによって設定された割り込み信号(この場合はIR10)が〝1〞になる．

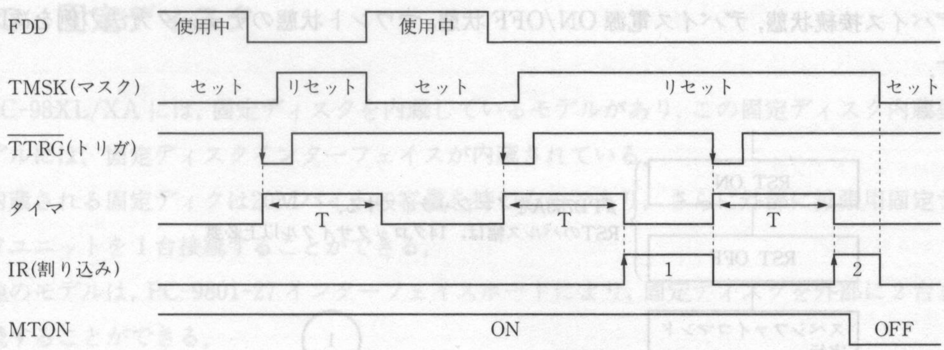
このタイマ機能には，トリガ入力後100ms以内に再びトリガを入力すれば，後のトリガが有効になるリトリガ機能を持つ．

ディスクドライブのモータをOFFにするためには，前記MTONビットを〝0〞にすればよい．次にモータON/OFF制御の具体例を示す．




注1：この手順に従うこと.
　　同時に設定も可.

注2：媒体セット時のみ.
　　RDYビットのチェックはセンスデバイスステータスのUS0／UL1セット直後がよい.




## (2) リードスイッチ／シグナル

●D7，D6ビット：FINT1，FINT0

640KBフロッピーディスクインターフェイス基板に搭載されているディップスイッチSW1のS7，S6ビット，割り込みレベル選択スイッチの状態を読み取る．システム既定値は，FINT1＝0，FINT0＝1(INT4＝IR10選択)となる．

●D5ビット：DMACH

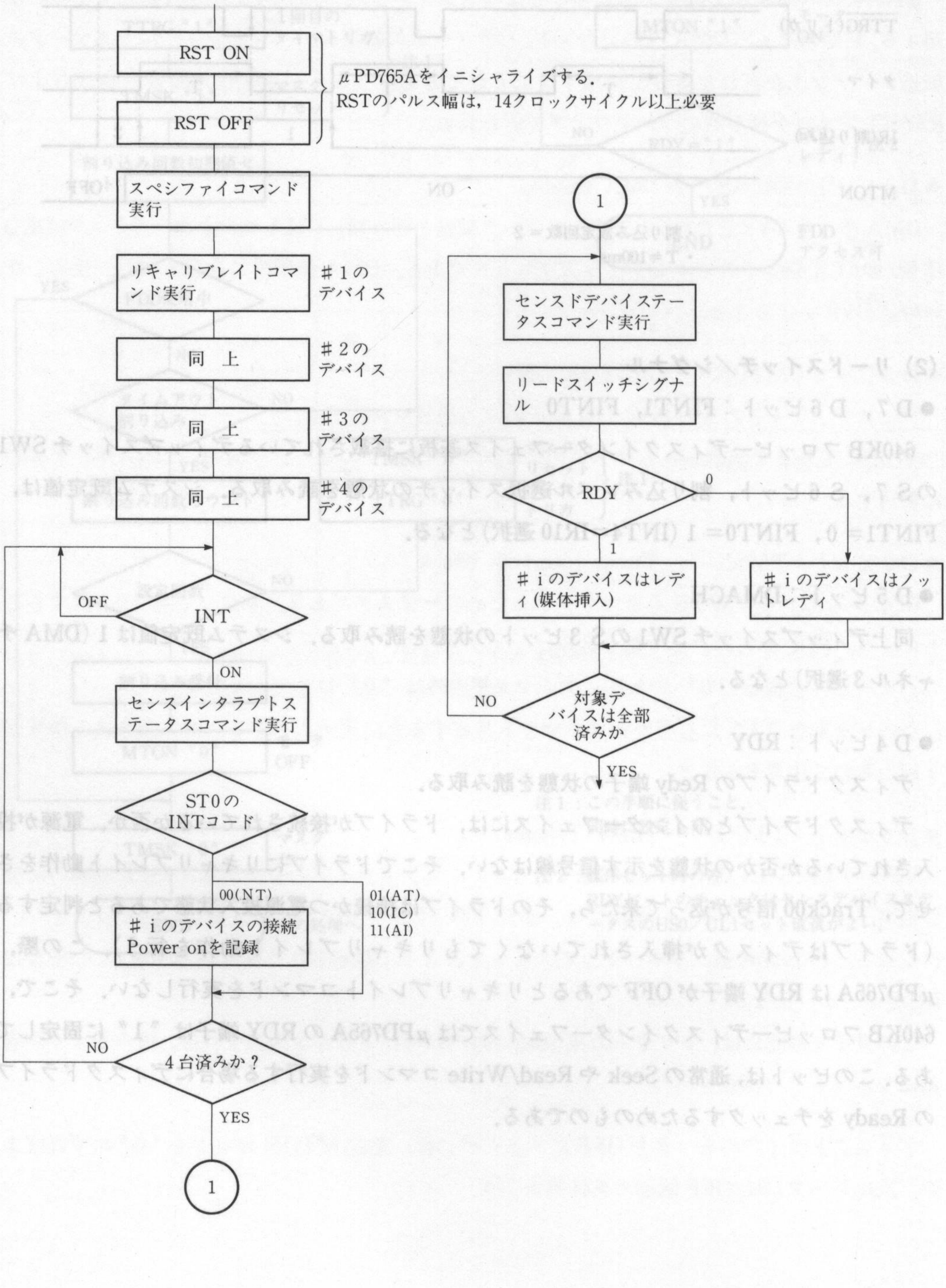
同上ディップスイッチSW1のS3ビットの状態を読み取る．システム既定値は1(DMAチャネル3選択)となる．

●D4ビット：RDY

ディスクドライブのRedy端子の状態を読み取る．

ディスクドライブとのインターフェイスには，ドライブが接続されているか否か，電源が投入されているか否かの状態を示す信号線はない．そこでドライブにリキャリブレイト動作をさせて，Track00信号が返って来たら，そのドライブは接続かつ電源投入状態であると判定する(ドライブはディスクが挿入されていなくてもリキャリブレイト動作を行う)．この際，μPD765AはRDY端子がOFFであるとリキャリブレイトコマンドを実行しない．そこで，640KBフロッピーディスクインターフェイスではμPD765AのRDY端子は〝1〟に固定してある．このビットは，通常のSeekやRead/Writeコマンドを実行する場合にディスクドライブのReadyをチェックするためのものである．

デバイス接続状態，デバイス電源ON/OFF状態，マウント状態のチェック方法の例を次図に示す．



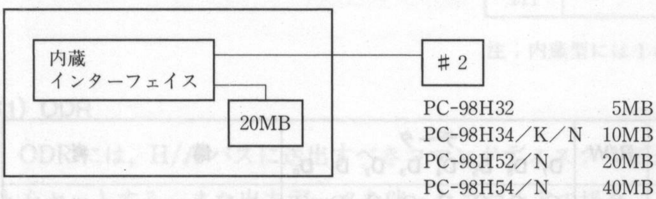
## 8.3　固定ディスク

PC-98XL/XA には，固定ディスクを内蔵しているモデルがあり，この固定ディスク内蔵型のモデルには，固定ディスクインターフェイスが内蔵されている．

内蔵される固定ディクは20Mバイトの容量を持つものであり，さらに外部に拡張用固定ディスクユニットを1台接続することができる．

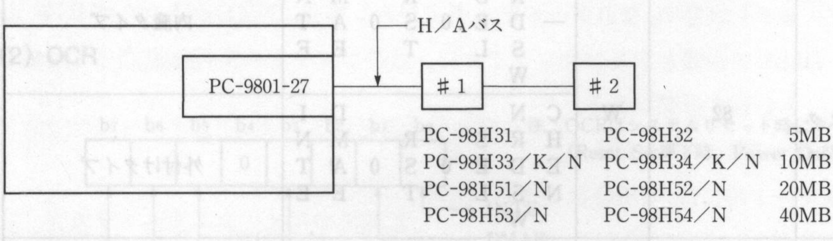
他のモデルは，PC-9801-27 インターフェイスボードにより，固定ディスクを外部に2台まで接続することができる．

●固定ディスク内蔵型




注：内蔵型インターフェイスは，以下の点が9801-27と異なっている．
・H／Aバスが外に出ていないため，1台目用の固定ディスクは接続できない．
・9800シリーズとはフォーマットが異なるため，9800シリーズで使用していた固定ディスクをそのまま接続してもデータは読み込めない(一度フォーマットしなおす必要がある)．

●固定ディスク外付け型




前記のいずれの場合にも，固定ディスクの増設を行う場合には，インターフェイスボード上のディップスイッチの再設定が必要である．

### (1) インターフェイスボードの仕様

| | PC-9801-27 | 内蔵型 |
|---|---|---|
| I/O モードアクセス | 下位バイト | |
| DMA モードアクセス | バイト転送 | |
| DMA チャネル | チャネル3 (XA), チャネル0 (XL) | |
| DMA 転送速度 | 196K～625K バイト/sec | 125K バイト/sec 以下 |
| 内蔵データバッファ | ドライブの種類により異なる | 約 8K バイト（トラックバッファ） |
| 割り込みレベル | INT3 (IR09) | |
| I/O ポートアドレス | 80h, 82h | |
| コントローラタイプ | 00 | 11 |

### (2) I/O アドレスと命令

| 命　令 | I/O ポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ODR アウトプットデータレジスタ | 80 | W | OD7 | OD6 | OD5 | OD4 | OD3 | OD2 | OD1 | OD0 | |
| IDR インプットデータレジスタ | 80 | R | ID7 | ID6 | ID5 | ID4 | ID3 | ID2 | ID1 | ID0 | |
| OCR アウトプットコントロールレジスタ | 82 | W | — | NRDSW | SEL | 0 | RST | 0 | DMAE | INTE | 内蔵タイプ |
| | | | CHEN | NRDSW | SEL | 0 | RST | 0 | DMAE | INTE | 外付けタイプ |
| ISR インプットステータスレジスタ | 82 | R | REQ | 0 | BSY | MSG | CXD | IXO | — | INT | 内蔵タイプ NRDSW=1 |
| | | | REQ | ACK | BSY | MSG | CXD | IXO | — | INT | 外付けタイプ NRDSW=1 |
| | | | CT1 | CT0 | DT02 | DT01 | DT00 | DT12 | DT11 | DT10 | NRDSW=0 |

固定ディスクインターフェイスのボード上には，接続される固定ディスクの種別を設定するディップスイッチがあり，OCRのNRDSWビットに0を書き込むことにより，ISRから読み取ることができる．

CT1, 0：コントローラタイプ
　　PC-9801-27　　＝00
　　内蔵インターフェイス＝11

DTxx　：デバイスタイプ
　　DT1x-1台目
　　DT0x-2台目

| デバイスタイプ | | 1台目 | 2台目 |
|---|---|---|---|
| 000 | 5MB | 98H31 | 98H32 |
| 001 | 10MB | 98H33/K/N<br>98H81 | 98H34/K/N |
| 100 | 20MB | 98H51/N<br>内蔵型 | 98H52/N |
| 110 | 40MB | 98H53/N | 98H54/N |
| 111 | | 未接続 | |

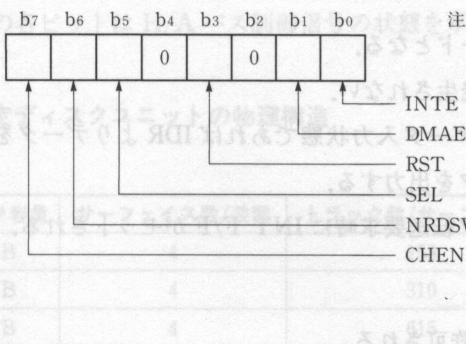
注：内蔵型には1台目用固定ディスクは接続できない．

(1) ODR

ODRには，H/Aバスに送出すべきコマンドディスクリプタブロック(CDB)のコマンドバイトをセットする．また出力データを伴ったコマンドの場合，出力データもこのポートを経由してH/Aバスに送出される．プログラムI/Oモード時には，H/Aが「データ出力状態」にあることを確認しながら，プログラムによりデータをセットする．DMAモード時には，H/Aが「データ出力状態」を検出し，データがDMA転送される．

選択状態に先だって，本ポートにコントローラ番号(01H)をセットすること．

(2) OCR




注：OCRはシステムリセット時にクリアされる．
(Reset Sw押下時，Power On時)

● CHEN : CHannel ENable

CHEN を1にすると，H/A バスへのデータの出力が許可される．

内蔵型インターフェイスには存在しない．

● NRDSW : ReaD SWitch

NRDSW により ISR の各ビットが切換えられる．

● SEL : SELect

SEL が1のとき，H/A バスの SEL 信号が ON になる．

● RST : ReSeT

RST を1から0にしたとき，H/A バスの RST 信号が 400nSec のパルス幅で ON になる．

H/A 自身のリセットは行われない．

● DMAE : Direct Memory Access Enable

DMAE をセットすると「DMA モード」となる．

DMA モード時には，H/A バスのデータ出力状態またはデータ入力状態を検出すると，DRQ 信号をシステムバスに対してアクティブにする．このときあらかじめ DMAC(8237)のパラメータ設定が適切になされていない場合，システムストールに陥る可能性があるので注意が必要．

DMAC が所定数のデータ転送を終了し，TC 信号を出力すると，プログラムからはアクセスできない DMAE F/F がセットされ，以後の DRQ の発生は禁止される．

DMAE F/F がリセット状態で，なおデータ転送要求がある場合，INTE が1ならば INT F/F が ON になる．DMA を必要とするコマンドごとに，このコマンドの実行に先立って，必ず DMAE F/F をセットし直す必要がある．

DMAE が0のときはプログラム I/O モードとなる．

プログラム I/O モード時には，DRQ は発生されない．

プログラムにより H/A バスを監視し，データ入力状態であれば IDR よりデータを引き取り，データ出力状態であれば ODR にデータを出力する．

なお，INTE＝1の場合は，最初のデータ転送要求時に INT F/F がセットされる．

● INTE : INTerrupt Enable

この信号が1のとき，割り込みの発生が許可される．

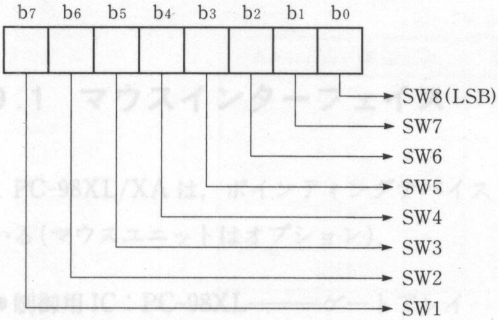
H/A バスのステータス状態を検出すると，INT F/F が ON になる．または，DMAE F/F リセット時には，データ入／出力状態を検出すると INT F/F が ON になる．

この信号が0のとき，INT F/F はリセット状態である．

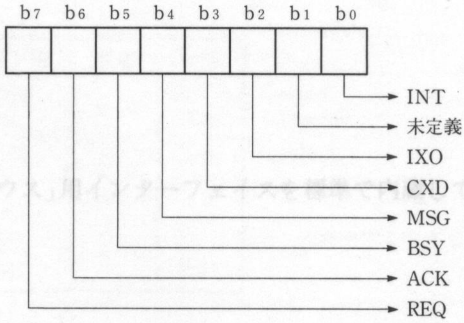
## (3) ISR

ISRの各ビットの意味は，OCRのNRDSWビットによって切換えられる．

・RDSW＝0のとき




・RDSW＝1のとき




● INT : INTerrupt

INT F/Fの状態を示す．INT F/FはOCRのINTEビットが0のときはリセット状態である．1のときInterrupt状態である．

● IXO : Input/$\overline{\text{Output}}$

● CXD : Command/$\overline{\text{Data}}$

● MSG : MeSsaGe

● BSY : BuSY

● ACK : ACknowledge

● REQ : REQuest

前記の各ビットはH/Aバス制御信号の状態を示す．

## (3) 固定ディスクユニットの物理構造

| ディスク容量 | サーフェイス数/装置 | トラック数/サーフェイス | セクタ数/トラック (注) | バイト数/セクタ (注) |
|---|---|---|---|---|
| 5MB | 4 | 153 | 33/17 | 256/512 |
| 10MB | 4 | 310 | 33/17 | 256/512 |
| 20MB | 4 | 615 | 33/17 | 256/512 |
| 40MB | 8 | 615 | 33/17 | 256/512 |

(注)左が外付け，右が本体内蔵

● CHEN : CHannel ENable

CHEN を1にする[illegible]
内蔵型インターフェイスには存在しない。

● NRDSW : ReaD SWitch

NRDSW によりISRの各ビットが切換えられる。

● SEL : SELect

SEL が1のとき、H/A バスの SEL 信号が ON になる。

● RST : ReSet

RST を1から0にしたとき、H/A バスの RST 信号が 400nSec のパルス幅で ON になる。
H/A 自身のリセットは行われない。

● DMAE : Direct Memory Access Enable

DMA モード時には、H/A バスのデータ出力状態または入力状態のとき、DRQ 信号をシステムバスに対してアクティブにする。このときあらかじめ DMAC のデータ設定が適切になされていない場合、システムストールに陥る可能性がある。

DMAC が所定数のデータ転送を終了し、TC 信号を出力すると、プログラム[illegible]できない DMAE F/F がセットされ、以後の DRQ の発生は禁止される。

DMAE F/F がリセット状態で、なおデータ転送要求がある場合、INT F/F が ON になる。DMA を必要とするコマンドごとに、このコマンドの実行前に DMAE F/F をセットし直す必要がある。

DMAE が0のときはプログラム I/O モードとなる。

プログラム I/O モード時には、DRQ は発生されない。

プログラムにより H/A バスを監視し、データ入力状態であれば IDR よりデータを引き取り、データ出力状態であれば ODR にデータを出力する。

● INTE : INTerrupt Enable

この信号が1のとき、割り込みの発生が許可される。

H/A バスのステータス状態を検出すると、INT F/F が ON になる。または、DMAE F/F リセット時には、データ入／出力状態を検出すると INT F/F が ON になる。

この信号が0のとき、INT F/F はリセット状態である。

# 第9章
# マウス

## 9.1 マウスインターフェイス

PC-98XL/XAは，ポインティングデバイス「マウス」用インターフェイスを標準で内蔵している(マウスユニットはオプション).

●制御用IC：PC-98XL———ゲートアレイ

PC-98XA———8255A

●ポート番号：

◇PC-98XA/XLハイレゾリューションモード

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | × | × | × | × | 0 | 0 | × | × | 0 | 1 | 1 | 0 | × | $A_1$ | $A_0$ | 1 | データポート |
| | × | × | × | × | 0 | 0 | × | × | 0 | 1 | 1 | 0 | × | $A_1$ | $A_0$ | 1 | 初期化割り込みタイミングのセット用 |

◇ノーマルモード

| ビット | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | $A_1$ | $A_0$ | 1 | データポート |
| | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | $A_1$ | $A_0$ | 1 | 初期化割り込みタイミングのセット用 |

●割り込み周期：8.33ms(120Hz既定値)

PC-98XA/XLハイレゾリューションモード時は，8.33ms固定となる.

PC-98XLノーマルモード時は, I/Oポート0BFDBHに次の値を書き込むことにより割り込み時間を設定する.

| 値(16進) | TT<br>10 | 周波数 | 時間 |
|---|---|---|---|
| 00 | 00 | 120Hz | 8 mS |
| 01 | 01 | 60Hz | 16mS |
| 02 | 10 | 30Hz | 34mS |
| 03 | 11 | 15Hz | 67mS |

●割り込みレベル：

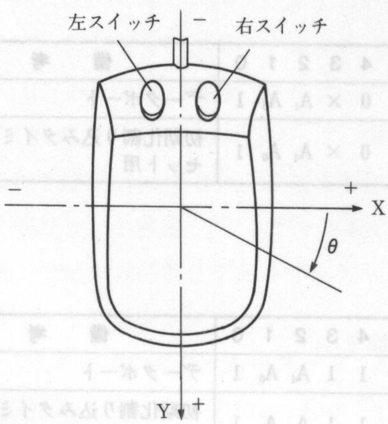
PC-98XA/XLハイレゾリューションモード時INT2固定.

PC-98XLノーマルモード時はINT6が既定値であるが，本体前面のスイッチカバー内ジャンパにより変更可能.

| ジャンパ番号 | 割り込み名 | 8259割り込みレベル |
|---|---|---|
| 1 | INT0 | IR 31 |
| 2 | INT1 | IR 51 |
| 3 | INT2 | IR 61 |
| 4 | INT3 | IR 91 |
| 5 | INT4 | IR101 |
| 6 | INT5 | IR121 |
| 7 | INT6 | IR131 |

●マウスの形状およびデータ

PC-9871，PC-9872



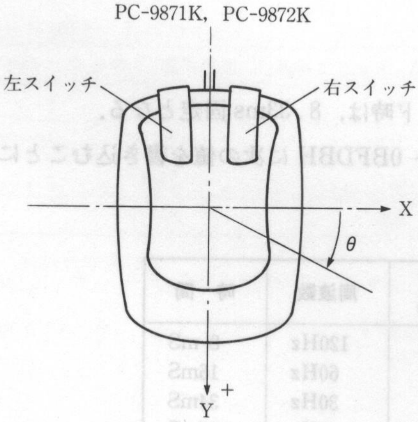

X＝A×L×cosθ±15〔カウント〕

Y＝A×L×sinθ±15〔カウント〕

A＝4〔カウント／mm〕

L＝移動距離〔mm〕

PC-9871K，PC-9872K



## 9.2　I/Oアドレスと命令

### (1) PC-98XA/XL ハイレゾリューションモード時

| 命　　令 | I／Oポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ライトモード | 0067 | W | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | モードセット |
| ライトポートC | 0067 | W | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0/1 | 割り込みEnable<br>0:Enable 1:Disable |
| | 0067 | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0/1 | Clear Count(HC)<br>0:クリアしない　1:クリアする |
| ライトポートC | 0065 | W | 0 | SXY | SHL | $\overline{INT}$ | 0 | 0 | 0 | 0 | ポートCはこの命令でも変更できる. |
| リードポートC | 0065 | R | HC | SXY | SHL | $\overline{INT}$ | MODSW | CPUSW | RS-2 | RS-1 | 下位4ビットによりスイッチの状態を読み取る. |
| リードポートB | 0063 | R | FDU | 1 | 1 | 1 | 1 | 1 | 1 | MOD81 | PC-98XL:外付け1MBFDの番号指定，8/10MHz CLOCK 切り替え<br>PC-98XA:8極ディップスイッチ(メモリサイズ) |
| リードポートA | 0061 | R | LEFT | × | RIGHT | × | MD3 | MD2 | MD1 | MD0 | マウスの状態を読み取る. |

HC　：0から1になったとき，その時のカウンタの値が読み出しバッファにコピーされた後クリアされ，新たにカウントを開始する.

$\overline{INT}$：0でマウスのタイマ割り込みを許可する

SXY, SHL：MD 3～0に出力するデータを選択する

| SXY | SHL | デ　ー　タ |
|---|---|---|
| 0 | 0 | X軸方向　下位4ビットデータ |
| 0 | 1 | 〃　　上位　〃 |
| 1 | 0 | Y軸方向　下位　〃 |
| 1 | 1 | 〃　　上位　〃 |

LEFT：0で左側のスイッチが押されている

RIGHT：0で右側のスイッチが押されている

MD 3～0：SXY，SHLで選択されたマウス用カウンタのデータ

MODSW　：ノーマル／ハイレゾリューションモードの設定状態(PC-98XLのみ)

0＝ハイレゾリューションモード

1＝ノーマルモード

CPUSW　：ディップスイッチ3-8（80286/70116切り替え）の設定状態(PC-98XLのみ)

0＝80286

1＝70116

RS-2，RS-1：RS-232C同期モード設定状態

PC-98XL――ディップスイッチ1-6，1-5

PC-98XA――ディップスイッチ1-10，1-9

FDU　：外付け1MB FDのユニット番号を#1・#2にするか#3・#4にするかの切り替え信号(ディップスイッチ1-4：PC-98XLのみ).

1＝外付けFD　#3，#4（既定値）

0＝外付けFD　#1，#2

MOD81　：ハイレゾリューションモード時の8/10MHzの切り替え信号(PC-98XLのみ).

本フラグによりクロック切り替えスイッチをソフトセンスする.

1＝8MHz

0＝10MHz

(2) PC-98XL ノーマルモード時

| 命　　令 | I/Oポート<br>アドレス | R/W | データ<br>$D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ライトモード | 7FDF | W | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | モードセット |
| ライト<br>ポートC | 7FDF | W | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0/1 | 割り込み Enable<br>0:Enable 1:Disable |
| | 7FDF | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0/1 | Clear Count(HC)<br>0:クリアしない 1:クリアする |
| ライト<br>ポートC | 7FDD | W | HC | SXY | SHL | $\overline{INT}$ | 0 | 0 | 0 | 0 | ポートCはこの命令でも変更できる. |
| リード<br>ポートC | 7FDD | R | HC | SXY | SHL | $\overline{INT}$ | MODSW | CPUSW | RS-2 | RS-1 | 下位4ビットによりスイッチの状態を読み取る. |
| リードポートB | 7FDB | R | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | PC−9801Vとの互換 |
| リード<br>ポートA | 7FD9 | R | LEFT | × | RIGHT | × | MD3 | MD2 | MD1 | MD0 | マウスの状態を読み取る. |
| ライトタイマ | BFDB | W | 0 | 0 | 0 | 0 | 0 | 0 | T1 | T0 | 割り込みタイマを設定する. |

T1, T0　: 割り込み周期選択.

ポートCの下位4ビットは, ライトモードによりマウスモードを93Hに設定した場合にのみ読み出すことができる. PC-9801VF/VMでは, マウスのモードを'10010xxx'と定義していたため, 従来のマウスドライバを使用した場合に読み出せない.

MODSW ：ノーマル/ハイレゾリューションモードの設定

0＝ハイレゾリューションモード

1＝ノーマルモード

CPUSW ：ディップスイッチ3-8 (V30/8086の切り替え) の設定値 (PC-98XL のみ)

0＝[illegible]

1＝[illegible]

RS ：RS-232C 同期モード設定状態

PC-98XL—ディップスイッチ1-5, 1-6

PC-98XA—ディップスイッチ1-10, 1-9

FDU ：FD1 を #1, #2 にするか #3, #4 にするかの切り替え信号(ディップスイッチ SW1-4、PC-98XL のみ)。

1＝[illegible] FD #3, #4

0＝外付け FD [illegible]

MOD81 ：ハイレゾリューションモード時の8/10MHzの切り替え信号(PC-98XL のみ)。

本フラグによりクロック切り替えスイッチをソフトセンスする。

1＝8MHz

0＝10MHz

# 第10章

# プリンタ

## 10.1　プリンタインターフェイス

PC-98XL/XAではプリンタを制御するために，セントロニクス仕様のパラレルインターフェイスを標準で実装している．制御用ICとして$\mu$PD8255を使用し，データバス16ビットのうち下位8ビット(LDB001～LDB071)と接続している．

ポートAはデータ出力，ポートCはデータストローブ信号(DATA・STB)と割り込み要求信号を発生する．ポートBはプリンタ側からのステータス信号を受けるために使用している．

PC-98XAおよびPC-98XLのハイレゾリューションモードにおいては，フルセントロニクス仕様となっており，PC-98XLノーマルモードにおいては，セントロニクス準拠の仕様となっている．

フルセントロニクス／セントロニクス準拠の切り替えはソフトウェアにより行う．

セントロニクス準拠時に使用しない信号線は，クランプまたは無視される．

●8255モードセットコマンド

| | I/Oポートアドレス | コマンド |
|---|---|---|
| ノーマルモード | 46H | 82H |
| ハイレゾリューションモード | 46H | A2H |

## 10.2 I/Oアドレスと命令

### 10.2.1 PC-98XA/XLハイレゾリューションモード時

フルセントロニクスプリンタの接続が可能である.

ポートAはモード1出力である(ただし，ストローブはソフトウェアにより発生させる).

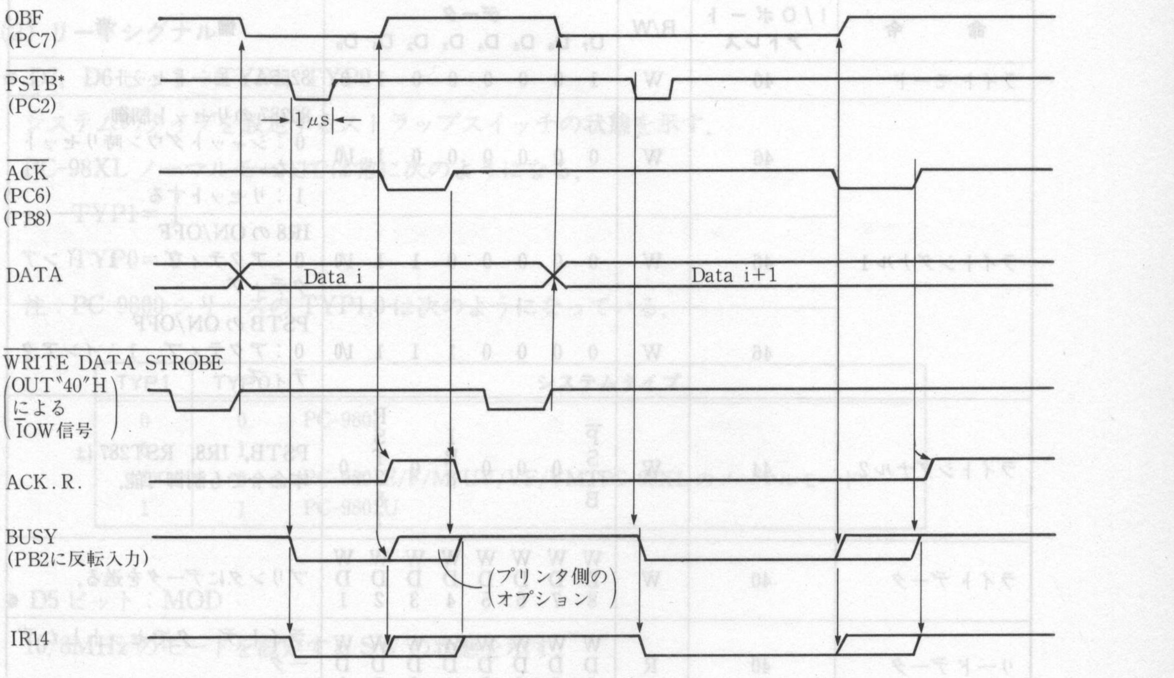
| 命　　令 | I/Oポートアドレス | R/W | データ $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ライトモード | 46 | W | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 8255A　モードセット |
| ライトシグナル1 | 46 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 | Input Prime ON/OFF<br>0：アクティブ　1：インアクティブ |
| | 46 | W | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1/0 | 80287のリセット制御<br>0：シャットダウン時リセットしない<br>1：リセットする. |
| | 46 | W | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1/0 | PSTB ON/OFF<br>0：アクティブ　1：インアクティブ |
| | 46 | W | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1/0 | INTE ON/OFF<br>0：禁止　　1：許可 |
| | 46 | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1/0 | OBF ON/OFF<br>0：アクティブ　1：インアクティブ |
| ライトシグナル2 | 44 | W | × | × | × | × | × | $\overline{\text{PSTB}}$ | RST287 | $\overline{\text{I}}$・PRIM | |
| ライトデータ | 40 | W | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | プリンタにデータを送る. |
| リード　データ<br>（診断用） | 40 | R | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | ライトデータでセットしたデータを読み込む. |
| リードシグナル1 | 42 | R | $\overline{\text{SELECT}}$ | $\overline{\text{FAULT}}$ | $\overline{\text{PE}}$ | $\overline{\text{DC5V}}$ | $\overline{\text{I}}$・BUSY | $\overline{\text{BUSY}}$ | $\overline{\text{ACK}}$・R | $\overline{\text{ACK}}$ | プリンタの状態を読み込む. |
| リードシグナル2 | 44 | R | $\overline{\text{OBF}}$ | $\overline{\text{ACK}}$ | × | × | IR14 | $\overline{\text{PSTB}}$ | $\overline{\text{RST287}}$ | $\overline{\text{I}}$・PRIM | 8255AのポートCの状態を読み込む. |

×：不定

↑（$D_5$）INTEの状態は見えない

### (1) 印字データ出力

プリンタ制御信号の内，$\overline{\text{PSTB}}$, ACK, INTR 信号は8255のポートAのモード1で規定されるコントロールポートで制御される．




＊PSTBはソフトウェアにより発生させる

### 10.2.2 PC-98XLノーマルモード時

割り込みの設定はソフトウェアにより行う必要があるので，有効利用はできない．

ポートCはモード0出力である．

| 命　令 | I/Oポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ライトモード | 46 | W | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 8255A　モードセット |
| ライトシグナル1 | 46 | W | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1/0 | 80287のリセット制御<br>0：シャットダウン時リセットしない<br>1：リセットする |
| | 46 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1/0 | IR8のON/OFF<br>0：アクティブ　1：インアクティブ |
| | 46 | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1/0 | PSTBのON/OFF<br>0：アクティブ　1：インアクティブ |
| ライトシグナル2 | 44 | W | $\overline{PSTB}$ | 0 | 0 | 0 | IR8 | 0 | RST287 | 0 | PSTB，IR8，RST287は本命令でも制御可能． |
| ライトデータ | 40 | W | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | プリンタにデータを送る． |
| リードデータ（診断用） | 40 | R | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | ライトデータでセットしたデータを読み込む． |
| リードシグナル1 | 42 | R | TYP1 | TYP0 | MOD | LCD | HGC | $\overline{BSY}$ | CPUT | 0 | プリンタの状態およびCPUのモード・タイプを読み込む． |
| リードシグナル2 | 44 | R | $\overline{PSTB}$ | × | × | × | IR8 | × | × | × | 8255AのポートCの状態を読み込む． |
| ライトポートC | 37 | W | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1/0 | PSTB信号Enable F/FのON/OFF<br>0：アクティブ　1：ノンアクティブ |

×　　：不定

IR8　：プリンタ制御回路から8259への割り込み信号

### (1) ライトシグナル

● D1：RST287

CPU リセット発生時，80287NDP をリセットするかどうかの指定.

このビットは 80286CPU 動作時のみ意味を持つ.

### (2) リードシグナル

● D7，D6 ビット：TYP1, TYP0

システムのタイプを設定するストラップスイッチの状態を示す.

PC-98XL ノーマルモードでは常に次のようになる.

TYP1＝1

TYP0＝0

注：PC-9800 シリーズの TYP1,0 は次のようになっている.

| TYP1 | TYP0 | システムタイプ |
|---|---|---|
| 0 | 0 | PC-9801 |
| 0 | 1 | |
| 1 | 0 | PC-9801E/F/M/UV/VF/VM, PC-98XL のノーマルモード |
| 1 | 1 | PC-9801U |

● D5 ビット：MOD

10/8MHz のモードを設定する SW の状態を示す.

0：10MHz

1：8MHz

このビットは，8/10MHz 切り替えスイッチの設定がそのまま見えるのではなく，80286CPU 動作時は，スイッチの設定に関係なく 8MHz に見える(スイッチの設定はマウスポート C から見える).

● D1 ビット：CPUT

動作 CPU の種別を示す.

0：80286

1：70116

このビットは，ディップスイッチ SW3-8 の設定がそのまま見えるのではなく，動作中の CPU に従った値となる(スイッチの設定はマウスポート C から見える).

### 10.2.2 PC-98XL ノーマルモード時

割り込みの設定はソフトウェアにより行う必要があるので，有効[illegible]

ポートCはモード0出力[illegible]

| 命令 | I/Oポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ライトモード | 46 | W | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | [illegible] |
| ライトシグナル1 | 46 | W | 0 | 0 | 0 | 0 | [illegible] | [illegible] | [illegible] | 1/0 | 80287リセット 0：シャットダウン時リセット 1：リセットする |
| | 46 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1/0 | IR8のON/OFF 0：アクティブ 1：インアクティブ |
| | 46 | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1/0 | PSTBのON/OFF 0：アクティブ 1：インアクティブ |
| ライトシグナル2 | 44 | W | PSTB | 0 | 0 | 0 | IR8 | [illegible] | RST | [illegible] | [illegible] |
| ライトデータ | 40 | W | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | プリンタにデータを送る。 |
| リードデータ（診断用） | 40 | R | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | [illegible]を読み込む。 |
| リードシグナル1 | 42 | R | TYP1 | TYP0 | M[illegible] | L[illegible] | H[illegible] | B[illegible] | CPU[illegible] | 0 | プリンタの状態とCPUの[illegible] |
| リードシグナル2 | 44 | R | [illegible]STB | × | × | × | [illegible]R8 | × | × | × | 8255AのポートCの状態を読み込む。 |
| ライトポートC | 37 | W | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1/0 | PSTB信号Enable，E/EのON/OFF 0：[illegible] |

×　：不定

# 第11章

# RS-232C

## 11.1 標準 RS-232C インターフェイス

### 11.1.1 概要

標準 RS-232C インターフェイスは制御部に μPD8251A を使用しており，調歩同期式全二重/半二重，同期式半二重のいずれか 1 チャンネルの通信が可能である．

通信速度は μPD8253 タイマに対する指示により与えられる．

| 通信方式 | 分　周　比 | 通　信　速　度（bps） |
|---|---|---|
| 同期式 | 1/1 | 75，150，300，600，1200，2400，4800，9600 |
| 調　歩 | 1/16 | 同　　上 |
| 同期式 | 1/64 | 75，150，300，600，1200，2400 |

また，同期方式の切換は本体前面のディップスイッチで行う．

RS-232C インターフェイスからの割り込みは TXRDY, TXEMPTY, RXRDY の 3 種の原因により発生し，各割り込みはシステムポート（μPD8255）の PC2, PC1, PC0（Interrupt Enable F/F）により制御が可能である．MODEM 信号のうち，CS, CD, CI はシステムポートの PB6, PB5, PB7 で読み取る事ができる．他の信号は μPD8251 のコントロール/ステータスで制御できる．

送受信データはデータバスの下位 8 ビットに乗せられる．

## 11.1.2　I/Oアドレスと命令

| 命　　令 | I/Oポートアドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| モード(A) | 32 | W | S2 | S1 | EP | PEN | $L_2$ | $L_1$ | $B_2$ | $B_1$ | $\mu$PD8251動作モードの設定（非同期） |
| モード(B) | 32 | W | SCS | ESD | EP | PEN | $L_2$ | $L_1$ | 0 | 0 | 〃（同期） |
| コマンド | 32 | W | EH | IR | RS | RST | SBR | REN | ER | TEN | |
| ステータス | 32 | R | DR | SYN | FE | OE | PE | TE | RRDY | TRDY | |
| データリード | 30 | R | RD8 | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | |
| データライト | 30 | W | SD8 | SD7 | SD6 | SD5 | SD4 | SD3 | SD2 | SD1 | |
| カウンタセット | 75 | W | $C_7$<br>$C_{15}$ | $C_6$<br>$C_{14}$ | $C_5$<br>$C_{13}$ | $C_4$<br>$C_{12}$ | $C_3$<br>$C_{11}$ | $C_2$<br>$C_{10}$ | $C_1$<br>$C_9$ | $C_0$<br>$C_8$ | |
| カウンタモード | 77 | W | SC1 | SC0 | RL1 | RL0 | $M_2$ | $M_1$ | $M_0$ | BCD | |
| マスクセット | 35 | W | × | × | × | × | × | TXRE | TXEE | RXRE | |
| リードシグナル | 33 | R | CI | CS | CD | × | × | × | × | × | システムポート |

×印：不定

## 11.1.3　μPD8251動作モードの設定

### (1) モードセット

調歩同期モード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| S2 | S1 | EP | PEN | L2 | L1 | B2 | B1 |

ボーレート

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 同期モード | ×1 | ×16 | ×64 |

キャラクタ長

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 5ビット | 6ビット | 7ビット | 8ビット |

パリティイネーブル
1：イネーブル
0：ディセーブル

偶数パリティ発生／チェック
1：偶数　0：奇数

ストップビット数

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 無　効 | 1ビット | 1.5ビット | 2ビット |

同期モード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| SCS | ESD | EP | PEN | L2 | L1 | 0 | 0 |

キャラクタ長

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 5ビット | 6ビット | 7ビット | 8ビット |

パリティイネーブル
1：イネーブル
0：ディセーブル

偶数パリティ発生／チェック
1：偶数　0：奇数

外部同期検出
1：SYNDET＝入力
0：SYNDET＝出力

単一キャラクタ同期
1：単一SYNCキャラクタ
0：ダブルSYNCキャラクタ

注：外部SYNCモードでは，二重キャラクタSYNCにプログラムすると，TXにのみ影響を与える．

### (2) コマンドセット

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| EH | IR | RTS | ER | SBRK | RXE | DTR | TxE |

- b0 TxE：送信イネーブル
  1：イネーブル　0：ディセーブル
- b1 DTR：データターミナルレディ　〝ハイレベル〟でDTR出力をゼロにする.
- b2 RXE：受信イネーブル
  1：イネーブル　0：ディセーブル
- b3 SBRK：センドブレイクキャラクタ
  1：TxD→0　0：通常動作
- b4 ER：エラーリセット
  1：すべてのエラーフラグ(PE, OE, FE)をリセット
- b5 RTS：センド要求　〝ハイレベル〟でRTS出力をゼロにする.
- b6 IR：内部リセット
  〝ハイレベル〟で8251Aをモードインストラクションフォーマットに戻す.
- b7 EH：HUNTモードへ
  1：SYNCキャラクタの検出を始める.
  *：アシンクロナス・モードでは無意味.

注：Rxイネーブルとエンターハントがプログラムされたらいつでもエラーリセットを実行する必要がある.
　　00Hを3回ダミーライトした後40Hをライトすると確実にリセットがかかる.

### (3) ステータスリード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| DSR | SYN DET | FE | OE | PE | TxE | Rx RDY | Tx RDY |

- b0 TxRDY：(注1)
- b6 SYNDET, b2 TxE, b1 RxRDY：外部入出力端子と同じ意味
- b3 PE：パリティエラー
  パリティエラーが検出されるとPEフラグがセットされる. これはコマンドインストラクションのERビットでリセットされる. PEは8251Aの動作を禁止しない.
- b4 OE：オーバーランエラー
  OEフラグはCPUがあるキャラクタを次のキャラクタが準備し終わる前に読み出さないときセットされる. これはコマンドインストラクションのERビットでリセットされる.
  OEは8251Aの動作を禁止しないが, 以前のオーバーランキャラクタは消滅する.
- b5 FE：フレーミングエラー(アシンクロナスモードのみ)
  FEフラグは各々の終りで有効ストップビットが検出されないときセットされる.
  これはコマンドインストラクションのERビットでリセットされる.
  FEは8251Aの動作を禁止しない.
- b7 DSR：データセットレディ
  DSRがゼロレベルにあることを示す.

注1：TxRDYステータスビットは, TxRDY出力端子と異った意味を持つ. ステータスビットはCTSとTxENに左右されず, 出力端子はCTSとTxENの両方に関連する.
　　TxRDYステータスビット＝DBバッファエンプティ
　　TxRDY出力端子＝DBバッファエンプティ(CTS＝0)・(TxET＝1)

## 11.2　拡張 RS-232C インターフェイス

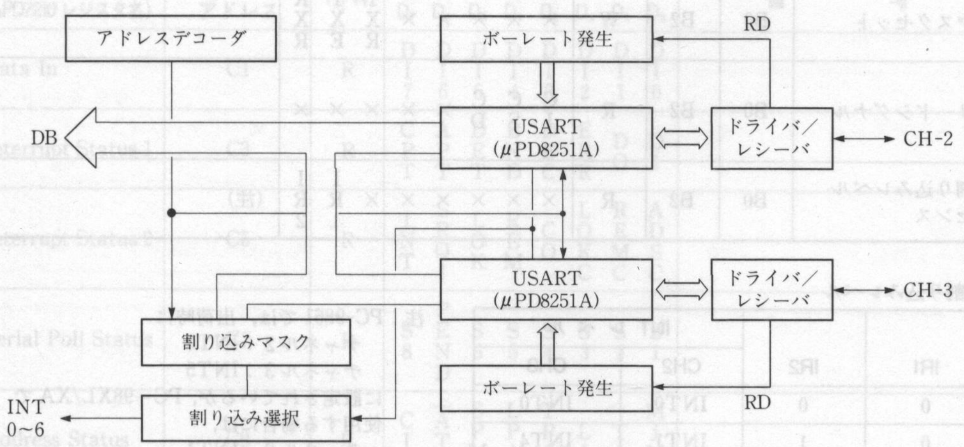
### 11.2.1　概要

PC-98XL/XA はオプションにより RS-232C インターフェイスを増設することができる．RS-232C インターフェイスを増設する場合，PC-9861/K(RS-232C 拡張(第 2 ～第 3 回線用)インターフェイスボード)を拡張スロットに実装する必要がある．ボード上の ROM は必ず KILL にしておくこと．

- ●チャネル数　　2CH
- ●ボーレート　　75～9600BPS(ディップスイッチによる設定)
- ● RS-232C インターフェイスは，制御部に μPD8251A 相当を使用

注：PC-9861 は，PC-98XL ノーマルモード 80286CPU 時，およびハイレゾリューションモード 10MHz 時は使用不可である．

### 11.2.2　ブロック図



### 11.2.3 I/Oアドレスと命令

| 命令 | I/Oポートアドレス CH2 | I/Oポートアドレス CH3 | R/W | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| モード(A) | B3 | BB | W | $S_2$ | $S_1$ | EP | PEN | $L_2$ | $L_1$ | × | × | μPD8251動作モードの設定<br>(非同期) |
| モード(B) | B3 | BB | W | SCS | ESD | EP | PEN | $L_2$ | $L_1$ | 0 | 0 | 〃<br>(同期) |
| コマンド | B3 | BB | W | EH | IR | RS | RST | SBR | REN | ER | TEN | |
| ステータス | B3 | BB | R | DR | SYN | FE | OE | PE | TE | RRDY | TRDY | |
| データリード | B1 | B9 | R | RD8 | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | |
| データライト | B1 | B9 | W | SD8 | SD7 | SD6 | SD5 | SD4 | SD3 | SD2 | SD1 | |
| マスクセット | B0 | B2 | W | × | × | × | × | × | TXR | TXE | RXR | |
| リードシグナル | B0 | B2 | R | CI | CS | CD | × | × | × | × | × | |
| 割り込みレベルセンス | B0 | B2 | R | × | × | × | × | × | × | IR1 | IR2 | (注) |

・割り込みレベル

| IR1 | IR2 | INTレベル CH2 | INTレベル CH3 |
|---|---|---|---|
| 0 | 0 | INT0 | INT0 |
| 0 | 1 | INT1 | INT4 |
| 1 | 0 | INT2 | INT5 |
| 1 | 1 | INT3 | INT6 |

注：PC-9861では，出荷時に
チャネル2：INT0
チャネル3：INT5
に設定されているが，PC−98XL/XAで使用する場合には，
チャネル2：INT1
チャネル3：INT0
を標準としているため，ボード上のディップスイッチを変更する必要がある．

# 第12章

# GP-IB

## 12.1 GP-IB インターフェイス

PC-98XL/XA には IEEE-488 のインターフェイスを持つ各種計測機器や周辺装置と接続できる GP-IB インターフェイスボードがオプションで提供されている(PC-9801-29/K).

インターフェイス LSI として μPD7210 が使用されている.

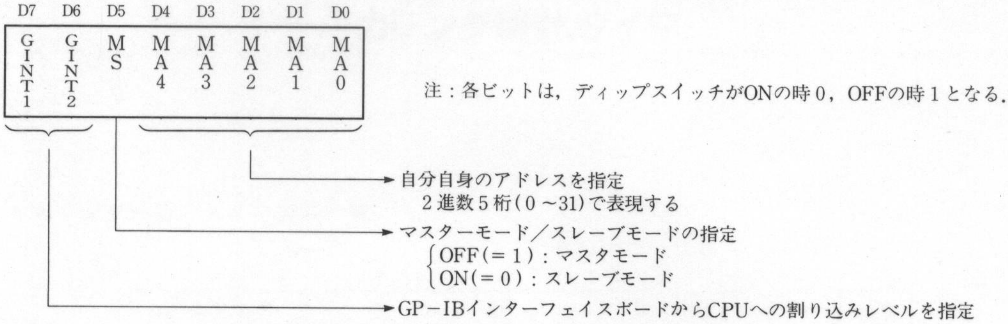
注：PC-9801-29 は，PC-98XL ノーマルモード 80286CPU 時は使用不可である.

## 12.2 I/O アドレスと命令

| 命令 (μPD7210 レジスタ名) | I/O ポートアドレス | R/W | データ $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Data In | C1 | R | DI7 | DI6 | DI5 | DI4 | DI3 | DI2 | DI1 | DI0 | |
| Interrupt Status 1 | C3 | R | CPT | APT | DET | END | DEC | ERR | DO | DI | |
| Interrupt Status 2 | C5 | R | INT | SRQI | LOK | REM | CO | LOKC | REMC | ADSC | |
| Serial Poll Status | C7 | R | S8 | PEND | S6 | S5 | S4 | S3 | S2 | S1 | |
| Address Status | C9 | R | CIC | $\overline{ATN}$ | SPMS | LPAS | TPAS | LA | TA | MJMN | |
| Command Pass Through | CB | R | CPT7 | CPT6 | CPT5 | CPT4 | CPT3 | CPT2 | CPT1 | CPT0 | |
| Address 0 | CD | R | X | DT0 | DL0 | AD5-0 | AD4-0 | AD3-0 | AD2-0 | AD1-0 | |
| Address 1 | CF | R | EOI | DT1 | DL1 | AD5-0 | AD4-0 | AD3-0 | AD2-0 | AD1-0 | |

| 命　　令 (μPD7210 レジスタ名) | I/Oポート アドレス | R/W | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | 備　　考 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte Out | C1 | W | BO7 | BO6 | BO5 | BO4 | BO3 | BO2 | BO1 | BO0 | |
| Interrupt Mask 1 | C3 | W | CPT | APT | DET | END | DEC | ERR | DO | DI | |
| Interrupt Mask 2 | C5 | W | 0 | SRQI | DMAO | DMAI | CO | LOKC | REMC | ADSC | |
| Serial Poll Mode | C7 | W | S8 | rsv | S6 | S5 | S4 | S3 | S2 | S1 | |
| Address Mode | C9 | W | ton | lon | TRM1 | TRM0 | 0 | 0 | ADM1 | ADM0 | |
| Auxiliary Mode | CB | W | CNT2 | CNT1 | CNT0 | COM4 | COM3 | COM2 | COM1 | COM0 | |
| Address 0/1 | CD | W | ARS | DT | DL | AD5 | AD4 | AD3 | AD2 | AD1 | |
| End of String | CF | W | EC7 | EC6 | EC5 | EC4 | EC3 | EC2 | EC1 | EC0 | |
| リードスイッチ | 99 | R | GINT1 | GINT2 | MS | MA4 | MA3 | MA2 | MA1 | MA0 | ボード上のスイッチ |
| リードIFC | 9B | R | $\overline{\text{IFC}}$ | GINT2 | MS | MA4 | MA3 | MA2 | MA1 | MA0 | ボード上のレジスタ |

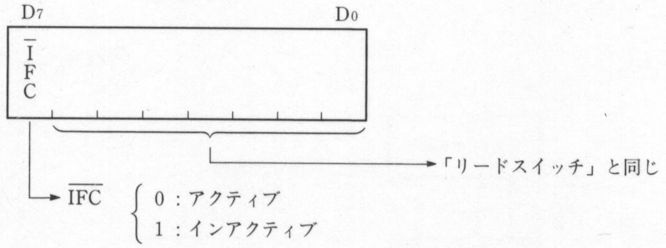
### (1) リードスイッチ(I/O アドレス 99)におけるデータビットの意味




| GINT1 | GINT2 | 割り込み名 | バススロット信号名 |
|---|---|---|---|
| 0 | 0 | INT 0 | IR 31 |
| 0 | 1 | INT 4 | IR 101 |
| 1 | 0 | INT 5 | IR 121 |
| 1 | 1 | INT 6 | IR 131 |

←システム既定値（INT 5 / IR 121）

### (2) リード IFC




### (3) 初期設定

① ステート移行禁止時間の設定

$N_F = 5$ ($F_3$　$F_2$　$F_1$　$F_0$＝0101)

② Aux.(B)レジスタ

$B_2 = 1$　ひき続くデータ送出の2バイト目以降のハンドシェイクの$T_1$として「$T_1$(高速)」を用いる.

$B_2 = 0$　INT 端子のアクティブレベルとして,「INT(アクティブハイ)」を指定.

③ T/R2, T/R3(トランスミット/レシーブ制御2，3)端子の機能選択

TRM0＝1
TRM1＝1 } ⇒ T/R2＝CIC
T/R3＝PE

### (1) リードスイッチ(I/Oアドレス99)におけるデータビットの意味

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| GINT1 | GINT2 | MS | MA4 | MA3 | MA2 | MA1 | MA0 |

- MA4～MA0 → 自分自身のアドレスを指定。2進数5桁(0～31)で表現する
- MS → マスターモード/スレーブモードの指定 { OFF(＝1)：マスターモード / ON(＝0)：スレーブモード }
- GINT1, GINT2 → GP-IBインターフェイスボードからCPUへの割り込みレベルを指定

注．各ビットは，ディップスイッチがONの時0，OFFの時1となる．

| GINT1 | GINT2 | 割り込み名 | バスコネクト信号名 |
|---|---|---|---|
| 0 | 0 | INT 2 | IR 11 |
| 0 | 1 | INT 4 | IR 101 |
| 1 | 0 | INT 3 | IR 121 |
| 1 | 1 | INT 6 | IR 131 |

### (2) リードIFC

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| IFC | | | | | | | |

- D4～D0 → 「リードスイッチ」と同じ
- IFC → { 0：アクティブ / 1：インアクティブ }

### (3) 初期設定

① ステート移行禁止時間の設定

$N_F = 5\ (F_3\ F_2\ F_1\ F_0 = 0101)$

② Aux.(B)レジスタ

$B_2 = 1$　ひき続くデータ送出の2バイト目以降のハンドシェイクの$T_1$として「$T_1$(高速)」を用いる．

$B_3 = 0$　INT端子のアクティブレベルとして，「INT(アクティブハイ)」を指定．

③ T/R2，T/R3(トランスミット/レシーブ制御2，3)端子の機能選択

TRM0＝1 ⇒ T/R2＝CIC

TRM1＝1 ⇒ T/R3＝PE

# 第3部

# 基本入出力プログラム

# 第1章
# カレンダ時計,タイマ

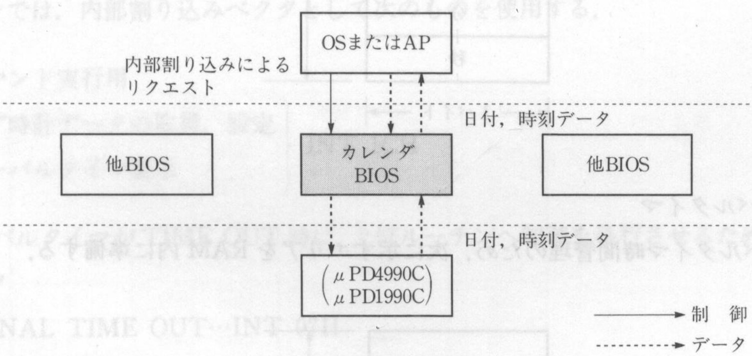
● カレンダ時計，タイマ機能一覧

| AH レジスタ | 機　能 | 備考 |
|---|---|---|
| 00H | 日付・時刻の読み出し | |
| 01H | 日付・時刻の設定 | |
| 02H | インターバルタイマ値の設定(シングルイベント) | |
| 03H | タイマキャンセル | 新規 |
| 04H | インターバルタイマ値の設定(ワンショットモード) | 新規 |
| 05H | インターバルタイマ値の設定(リピートモード) | 新規 |
| 06H | ビープ機能 | 新規 |

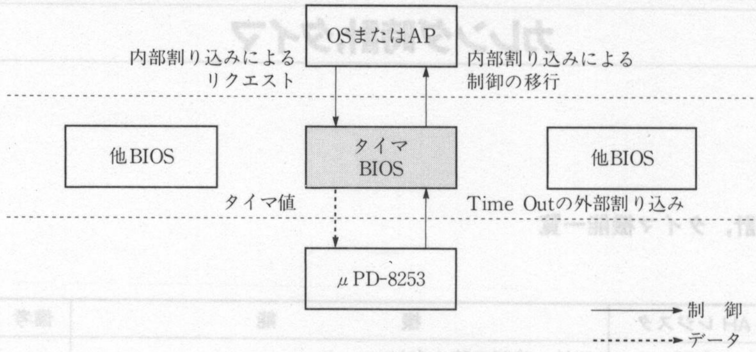
## 1.1 カレンダ時計，タイマBIOS 共通情報

### (1) 本ルーチンの位置づけ

① カレンダ時計



② インターバルタイマ

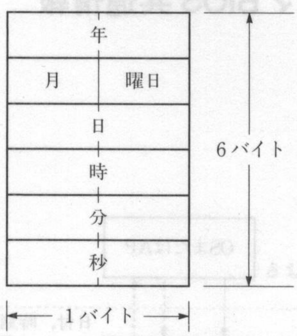



上位モジュールとのインターフェイスは，機能コードを伴なった内部割り込みによる．

### (2) モジュール間インターフェイスRAMエリア

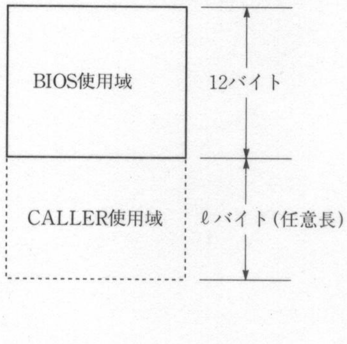
① カレンダ時計

カレンダデータ受け渡しのため，次に示すエリアをRAM内に準備する．




② インターバルタイマ

インターバルタイマ時間管理のため，次に示すエリアをRAM内に準備する．



### (3) システム共通域

① 本ルーチンではローカルRAMエリアとして「年」格納のために不揮発メモリ１バイト使用する．

② インターバルタイマ，キュー用のルートリストポインタとして４バイト，PC-9800シリーズ互換タイマ値格納域として２バイト，上位ルーチンのアドレスとして４バイト，Beepタイマ値格納域として２バイト，音調用として２バイトを使用する．

| システム共通域<br>シンボル名・絶対番地 | 用　　途 |
|---|---|
| CAL_ROOT_LIST<br>(444H～447H) | インターバルタイマ，マルチイベントのパラメータブロックの待ち行列の先頭を指すポインタ． |
| CAL_BEEP_TIME<br>(448H～449H) | インターバルタイマ，Beep機能のBeepの継続時間．(10msec単位) |
| CAL_TONE<br>(44AH～44BH) | Beepの音調 |
| CAL_USER_OFF<br>(44CH～44DH) | シングルイベントタイムアウト時のユーザールーチンのオフセットアドレス． |
| CAL_USER_SEG<br>(44EH～44FH) | シングルイベントタイムアウト時のユーザールーチンのセグメントアドレス． |
| CAL_SINGLE<br>(58AH～58BH) | シングルイベントのタイマ値格納域． |

### (4) 内部割り込みベクタ

本ルーチンでは，内部割り込みベクタとして次のものを使用する．

① BIOSコマンド実行用

・カレンダ時計データの取得，設定<br>
・インターバルタイマ要求　} INT 1CH

② インターバルタイマがTIME OUT時に，上位ルーチンへ制御を移行させるための内部割り込みベクタ

・INTERNAL TIME OUT…INT 07H

## 1.2　日付・時刻の読み出し

### (1) 機能

現在の日付(年，月，曜日，日)，時刻(時，分，秒)の読み出しを行う．

### (2) 入力

・内部割り込みコード←1CH

・AH ← 00H

・ES/BX ←日付，時刻の通知を受けるデータバッファ(6バイト)のアドレスを指定
　(ES ←セグメントベース，BX ←オフセット)

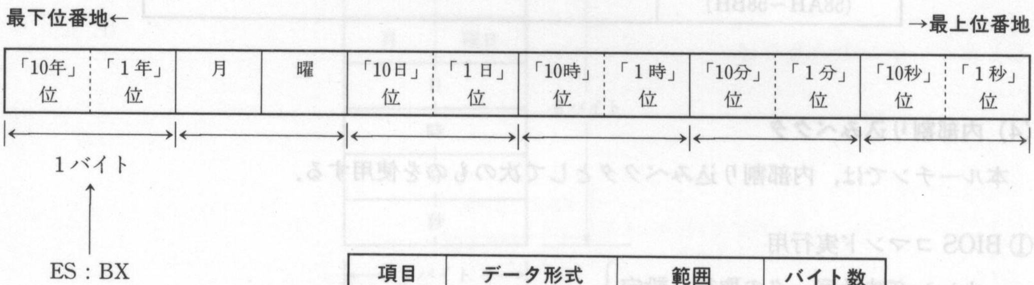
### (3) 出力

AX 以外のレジスタはすべて保証される．

### (4) 処理

現在の年，月，曜日，日，時，分，秒を $\mu$PD4990C より読み出して，データバッファに格納する．

### (5) データバッファ形式




| 項目 | データ形式 | 範囲 | バイト数 |
|---|---|---|---|
| 年 | BCD | 00H～99H | 1 |
| 月 | 16進数 | 01H～0CH | 1 |
| 曜 | 16進数 | 00H～06H | |
| 日 | BCD | 01H～31H | 1 |
| 時 | BCD | 00H～23H | 1 |
| 分 | BCD | 00H～59H | 1 |
| 秒 | BCD | 00H～59H | 1 |

### (6) 注意

・PC-98XA は，「年」は不揮発性メモリに格納されるために，自動的に更新されない．また，うるう年もサポートされない．

・「月」は月の大小を自動判別する．

## 1.3 日付・時刻の設定

### (1) 機能

システムの日付を設定する．

### (2) 入力

・内部割り込みコード←1CH

・AH←01H

・ES/BX←設定する日付・時刻が格納されているデータバッファ(6バイト)のアドレスを指定(ES←セグメントベース，BX←オフセット)

### (3) 出力

AX以外のレジスタはすべて保証される．

### (4) 処理

データバッファに格納されている年，月，曜，日，時，分，秒を，μPD4990C(XAはμPD1990C)へ設定し，時刻の動作を開始する．

### (5) データバッファ形式

最下位番地← →最上位番地

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 「10年」位 | 「1年」位 | 月 | 曜 | 「10日」位 | 「1日」位 | 「10時」位 | 「1時」位 | 「10分」位 | 「1分」位 | 「10秒」位 | 「1秒」位 |

1バイト

↑

ES：BX

| 項目 | データ形式 | 範囲 | バイト数 |
|---|---|---|---|
| 年 | BCD | 00H～99H | 1 |
| 月 | 16進数 | 01H～0CH | 1 |
| 曜 | 16進数 | 00H～06H | |
| 日 | BCD | 01H～31H | 1 |
| 時 | BCD | 00H～23H | 1 |
| 分 | BCD | 00H～59H | 1 |
| 秒 | BCD | 00H～59H | 1 |

### (6) 注意

1.2(6)参照

## 1.4　インターバルタイマ

インターバルタイマは次の機能を持つ.

・インターバルタイマ値を設定し，インターバルタイマを起動する．設定値まで時間が経過するとタイムアウトになり，タイマ割り込みが発生し，戻り番地に制御を移す.

・マルチイベントのインターバルタイマ要求を処理することが可能である．上位はあたかも複数のタイマが存在するかのようにタイマ要求を発行できる．たとえば連続して，10msec, 50msec, 100msec 要求を発行し，各々のタイムアウト時，各々異なるイベントを発生させることも可能である.

・リピーテッドモードを提供する．このモードでタイマ要求を行うと，タイムアウトになりイベント発生後，再度同じ値でタイマ要求を行う．つまり，一度このモードで要求すると，繰り返してインターバルタイマ要求が発行される.

### (1) 処理ルーチンの記述

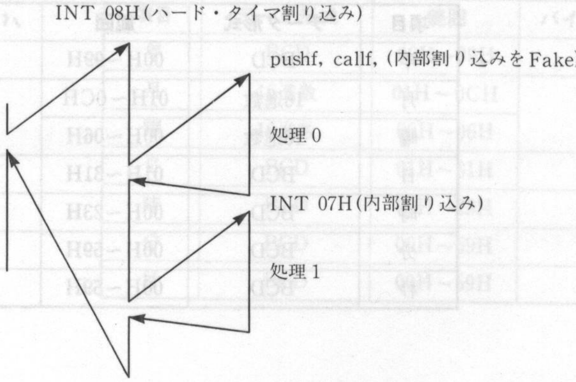
・処理ルーチンは，タイマ割り込み処理ルーチンより，マルチイベントならINT 07H の内部割り込みで，シングルイベントの場合は Pushf と Callf を使って内部割り込みと同じレベルで呼び出される.

・処理ルーチンは, AX 以外のすべてのレジスタを保証しなければならない．タイマ割り込み処理ルーチンへは IRET で返る.

・リターンコードとして, AH＝00 または FFH を返す．リピーテッドモード時には，このリターンコードで次の処理が決定される.

・制御の移行

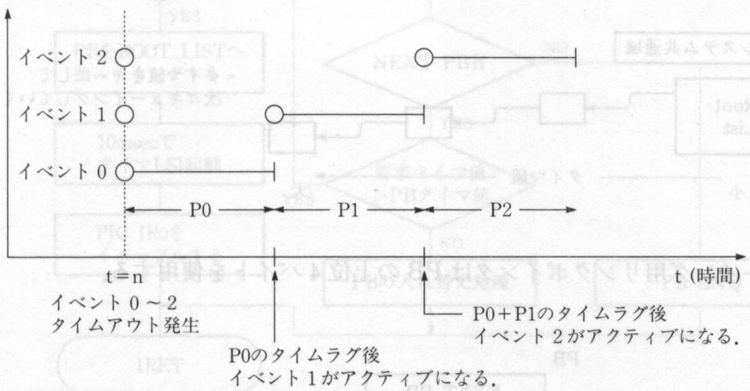
［例］



### (2) インターバルタイマ使用留意点

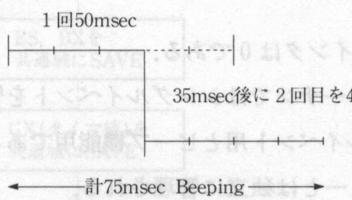
マルチイベントモード使用時，同時に複数のタイムアウトが発生した場合，先行イベント処理終了後に後発イベントがアクティブになるため，先行イベント処理時間分だけタイムラグが生じる．

シングルイベントタイムアウト時も，処理時間分だけ後発イベントが遅れる．上記のタイムラグは積算的なので注意を要する．




Beepで要求時間経過前に次のビープ要求が発行さると，本ルーチンは次に示すように対処する．

［例］

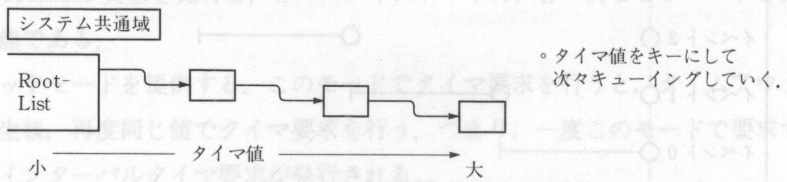


### (3) マルチイベントタイマの実現方式

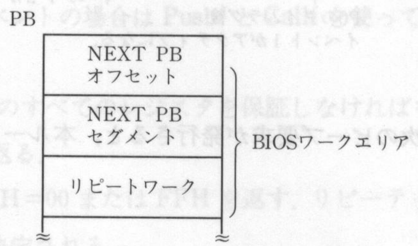
本ルーチンは上位からのタイマ要求を処理するタイマ要求処理モジュールと，10 msec 間隔で発生するタイマ割り込みを処理する割り込み処理モジュールより構成される．

a）タイマ要求処理モジュール

・現在与えられたタイマ値と，すでにキューイングされている PB 群のタイマ値を比較して，タイマ値の小さいものから順にキューイングされているようにする．




・キューイング用リンクポインタは PB の上位 4 バイトを使用する．




・NEXT PB が存在しない時，このポインタは 0 である．

・マルチイベント管理に加えて，このルーチンではシングルイベントを管理する 2 つのカウンタを使っている．通常のシングルイベント用とビープ機能用である．これらのカウンタは，マルチイベントの PB のキューとは独立に管理される．

b）割り込み処理モジュール

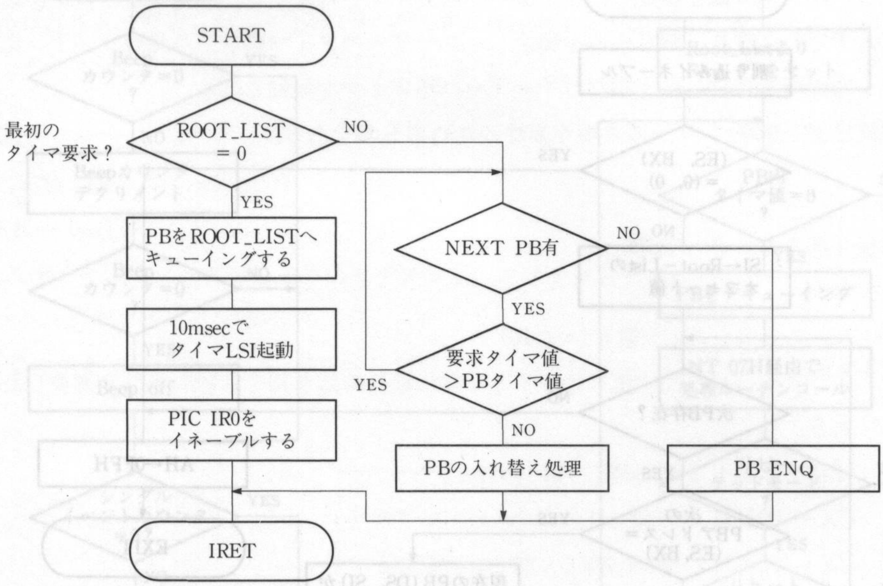
・10msec 毎にタイマ割り込みが発生する．ビープ機能のカウンタをデクリメントする．0 になったらビープを off にする．

・シングルイベントのカウンタをデクリメントする．0 になったら当該ルーチンをコールする (Pushf，Callf を使って INT を fake する)．

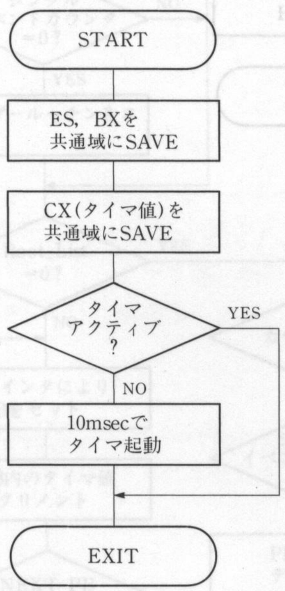
・次に Root-List よりタイマ格納域が 0 の PB に対して順に INT 07H の内部割り込みを使用して，PB のデキュー後制御を渡す．

c）概要フロー

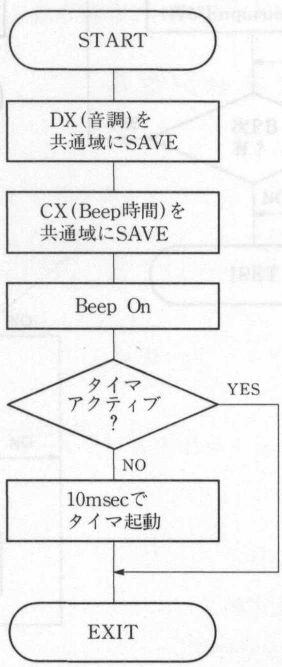
① タイマ要求処理



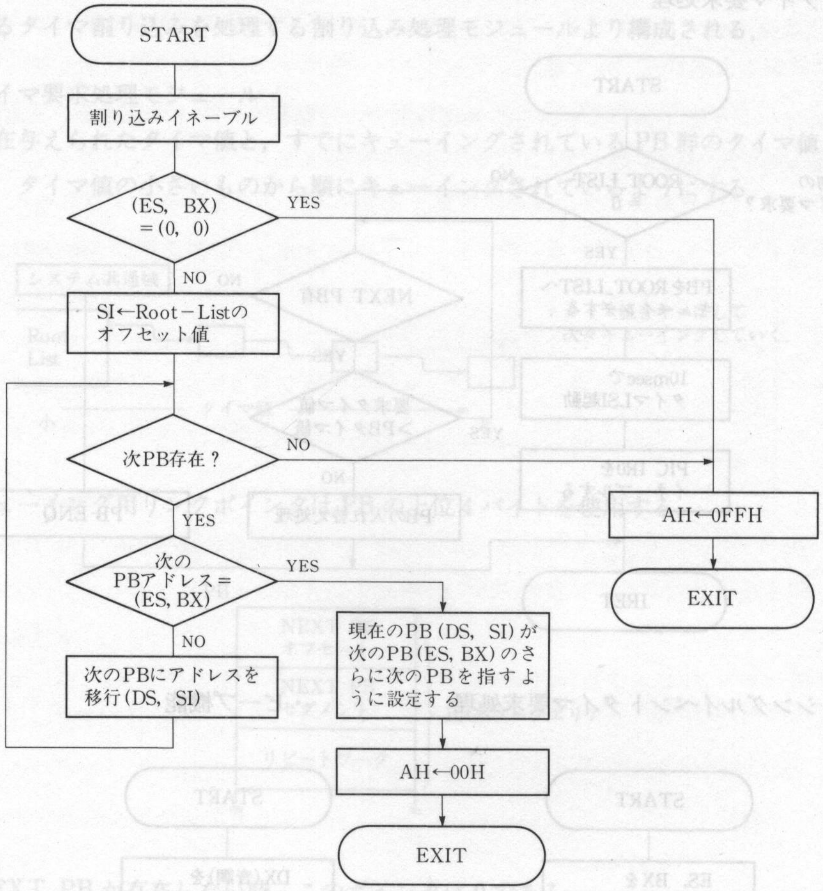

・シングルイベントタイマ要求処理



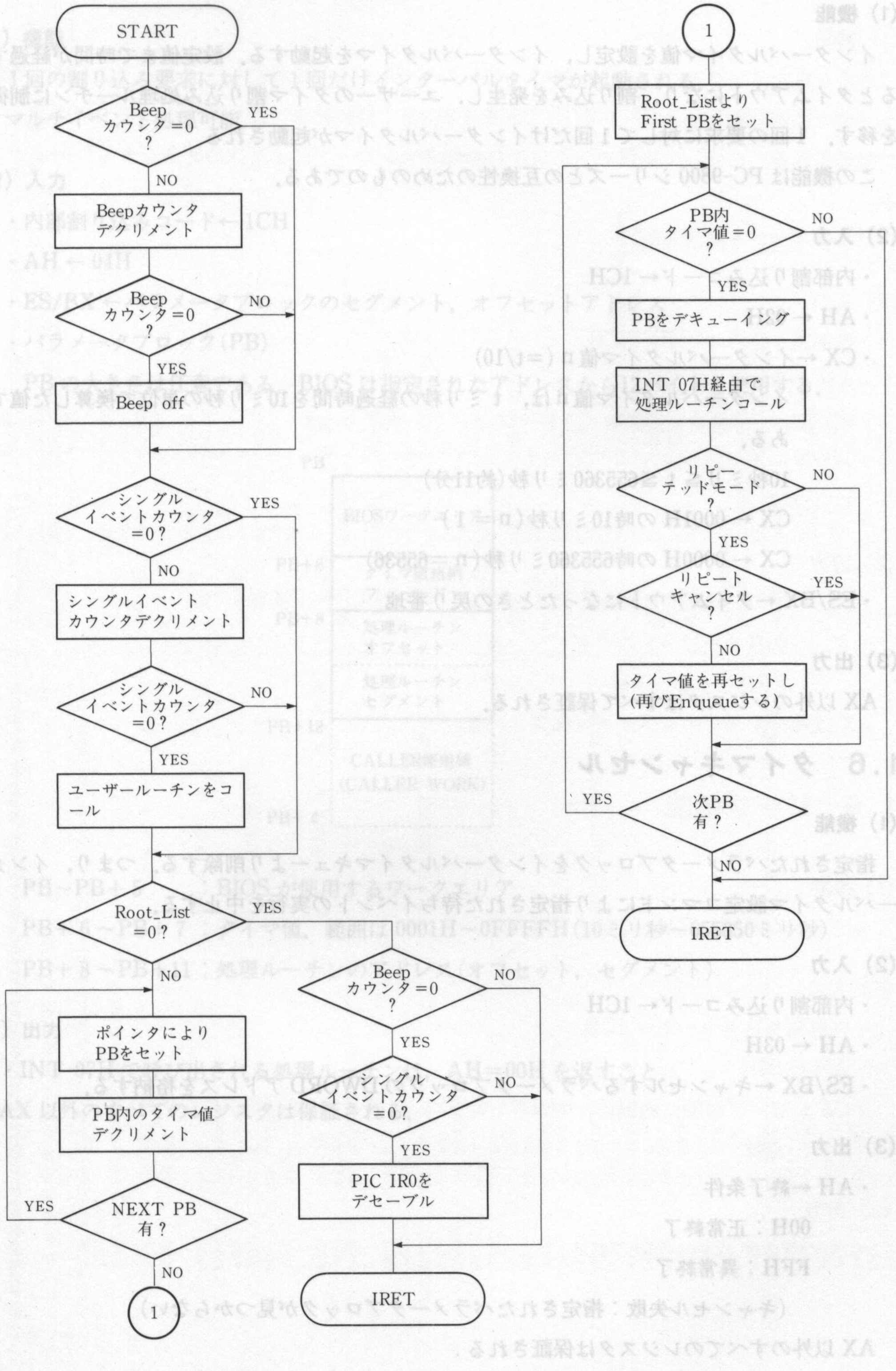

・ビープ機能



・インターバルタイマキャンセル ルーチンフロー



② 割り込み処理


START
Beep
カウンタ＝0
?
YES
NO
Beepカウンタ
デクリメント
Beep
カウンタ＝0
?
NO
YES
Beep off
シングル
イベントカウンタ
＝0？
YES
NO
シングルイベント
カウンタデクリメント
シングル
イベントカウンタ
＝0？
NO
YES
ユーザールーチンをコール
Root_List
＝0？
YES
NO
ポインタにより
PBをセット
PB内のタイマ値
デクリメント
YES
NEXT PB
有？
NO
1
Beep
カウンタ＝0
?
NO
YES
シングル
イベントカウンタ
＝0？
NO
YES
PIC IR0を
デセーブル
IRET
1
Root_Listより
First PBをセット
PB内
タイマ値＝0
?
NO
YES
PBをデキューイング
INT 07H経由で
処理ルーチンコール
リピー
テッドモード
?
NO
YES
リピート
キャンセル
?
YES
NO
タイマ値を再セットし
(再びEnqueueする)
YES
次PB
有？
NO
IRET

## 1.5　インターバルタイマ値の設定(シングルイベント)

### (1) 機能

インターバルタイマ値を設定し，インターバルタイマを起動する．設定値まで時間が経過するとタイムアウトになり，割り込みを発生し，ユーザーのタイマ割り込み処理ルーチンに制御を移す．1回の要求に対して1回だけインターバルタイマが起動される．

この機能はPC-9800シリーズとの互換性のためのものである．

### (2) 入力

・内部割り込みコード←1CH

・AH←02H

・CX←インターバルタイマ値n(=t/10)

インターバルタイマ値nは，tミリ秒の経過時間を10ミリ秒の単位で換算した値である．

10秒ミリ≦t≦655360ミリ秒(約11分)

CX←0001Hの時10ミリ秒(n=1)

CX←0000Hの時655360ミリ秒(n=65536)

・ES/BX←タイムアウトになったときの戻り番地

### (3) 出力

AX以外のレジスタはすべて保証される．

## 1.6　タイマキャンセル

### (1) 機能

指定されたパラメータブロックをインターバルタイマキューより削除する．つまり，インターバルタイマ設定コマンドにより指定された待ちイベントの実行を中止する．

### (2) 入力

・内部割り込みコード←1CH

・AH←03H

・ES/BX←キャンセルするパラメータブロックのDWORDアドレスを格納する．

### (3) 出力

・AH←終了条件

00H：正常終了

FFH：異常終了

(キャンセル失敗：指定されたパラメータブロックが見つからない)

AX以外のすべてのレジスタは保証される．

## 1.7 インターバルタイマ値の設定(ワンショットモード)

### (1) 機能

1回の割り込み要求に対して1回だけインターバルタイマが起動される．

マルチイベント処理可能．

### (2) 入力

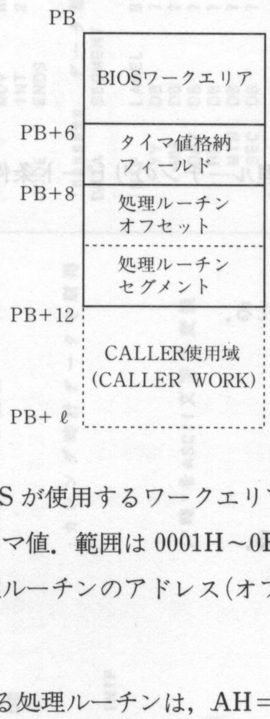
・内部割り込みコード← 1CH

・AH ← 04H

・ES/BX ←パラメータブロックのセグメント，オフセットアドレス

・パラメータブロック(PB)

PBの大きさは任意である．BIOSは指定されたアドレスから12バイトを使用する．




PB～PB＋5　　　：BIOSが使用するワークエリア

PB＋6～PB＋7：タイマ値．範囲は0001H～0FFFFH(10ミリ秒～655350ミリ秒)

PB＋8～PB＋11：処理ルーチンのアドレス(オフセット，セグメント)

### (3) 出力

・INT 07Hで呼び出される処理ルーチンは，AH＝00Hを返すこと．

AX以外のすべてのレジスタは保証される．

## 1.8　インターバルタイマ値の設定(リピートモード)

**(1) 機能**

与えられたインターバルタイマ値で，インターバルタイマを起動し，タイムアウト時に処理ルーチンに内部割り込み(INT 07H)で制御を渡す．1回の要求で繰り返しインターバルタイマが起動される．ただし，処理ルーチンからのリターンコードにより次の要求をキャンセルできる．

マルチイベント処理可能．

**(2) 入力**

・内部割り込みコード←1CH

・AH←05H

・ES/BX←パラメータブロックのセグメント，オフセット

・パラメータブロック

　1.7に同じ

**(3) 出力**

・AH←INT 07Hで呼び出される処理ルーチンのリピート条件

　　00H：リピートの中止

　　FFH：リピート続行

AX以外のすべてのレジスタは保証される．

## 1.9　ビープ機能

**(1) 機能**

与えられた時間だけビープ音を発生する．

**(2) 入力**

・内部割り込みコード←1CH

・AH←06H

・CX←ビープ時間．範囲は，0001H～0FFFFH(10ミリ秒～655350ミリ秒)

・DX←音調

**(3) 出力**

すべてのレジスタは保証される．

●**サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;##############################################################
;#                                                           #
;#       カレンダ時計のデータを取得する                        #
;#                                                           #
;##############################################################
CODE     SEGMENT
         ASSUME   CS:CODE,DS:DATA,SS:STACK
START:
         MOV      AX,SEG DATA               ;データセグメントを設定
         MOV      DS,AX
;
         MOV      AH,2                      ;"*"を表示
         MOV      DL,1AH
         INT      21H
         MOV      AH,9                      ;カーソルを消去
         MOV      DX,OFFSET DC
         INT      21H
;
         MOV      AX,DS                     ;カレンダ時計データを取得
         MOV      ES,AX
         MOV      BX,OFFSET INTF
TIMER:
         MOV      AH,Ø
         INT      1CH
T_HH:    MOV      AL,HH                     ;"時"をASCII文字に変換
         MOV      AH,AL
         AND      AH,ØFØH
         MOV      CL,4
         SHR      AH,CL
         OR       AH,3ØH
         MOV      ZH,AH
         AND      AL,ØFH
         OR       AL,3ØH
         MOV      ZL,AL
T_MM:    MOV      AL,MIN                    ;"分"をASCII文字に変換
         MOV      AH,AL
         AND      AH,ØFØH
         MOV      CL,4
         SHR      AH,CL
         OR       AH,3ØH
         MOV      MH,AH
         AND      AL,ØFH
         OR       AL,3ØH
         MOV      ML,AL
T_SS:    MOV      AL,SEC                    ;"秒"をASCII文字に変換
         MOV      AH,AL
         AND      AH,ØFØH
         MOV      CL,4
         SHR      AH,CL
         OR       AH,3ØH
         MOV      SH,AH
         AND      AL,ØFH
         OR       AL,3ØH
         MOV      SL,AL
DISPLAY:
         MOV      AH,9                      ;時刻を表示
         MOV      DX,OFFSET MSG
         INT      21H
;
         MOV      AH,ØBH                    ;キー入力待ち
         INT      21H
         CMP      AL,ØØH
         JE       TIMER
;
         MOV      AH,9                      ;カーソルを表示
         MOV      DX,OFFSET EC
         INT      21H
EXIT:
         MOV      AH,4CH                    ;プログラムを終了
         INT      21H
CODE     ENDS
;#######  データ領域
DATA     SEGMENT
;                                           インターフェイスエリア
INTF     LABEL    BYTE
  YY     DB       ?                         ;年
  MM     DB       ?                         ;月／曜日
  DAY    DB       ?                         ;日
  HH     DB       ?                         ;時
  MIN    DB       ?                         ;分
  SEC    DB       ?                         ;秒
;                                           メッセージ
MSG      DB       '現在の時刻は、'
  ZH     DB       ?
  ZL     DB       ?,'時'
  MH     DB       ?
  ML     DB       ?,'分'
  SH     DB       ?
  SL     DB       ?,'秒です．',ØDH,ØAH,ØAH
         DB       '何かキーを押すと終了します．',1EH,'$'
DC       DB       1BH,'[>5h','$'
EC       DB       1BH,'[>5l','$'
DATA     ENDS

;#######  スタック領域
STACK    SEGMENT  STACK
         DW       4ØH DUP (?)
STACK    ENDS

         END      START
```

```
;###################################################################
;#                                                                 #
;#        日付・時刻を設定する                                     #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AX,DS                   ;日付・時刻を設定
        MOV     ES,AX
        MOV     BX,OFFSET INTF
        MOV     AH,1
        INT     1CH
;
        MOV     AH,9                    ;設定日付・時刻を表示
        MOV     DX,OFFSET MSG
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  データ領域
DATA    SEGMENT
;
INTF    LABEL   BYTE                    ;インターフェイスエリア
  YEAR  DB      88H                       ;年
  MONTH DB      11H                       ;月／曜日
  DAY   DB      Ø1H                       ;日
  HH    DB      12H                       ;時
  MM    DB      3ØH                       ;分
  SEC   DB      45H                       ;秒
;
MSG     DB      1AH,'日付・時刻を「１９８８年０１月０１日'
        DB      '12時３０分４５秒に設定しました．$'
DATA    ENDS
;#########  スタック領域
STACK           SEGMENT STACK
                DW      8ØH DUP (?)
STACK           ENDS
                END     START
```

```
;###################################################################
;#                                                                 #
;#        インターバルタイマーを設定する                           #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,9                    ;割込待ちメッセージを表示
        MOV     DX,OFFSET MSG_W
        INT     21H
;
        MOV     AX,INTR                 ;タイマーを設定
        MOV     ES,AX
        MOV     BX,OFFSET I_ENT
        MOV     CX,5ØØ                            ;割込間隔 =5秒
        MOV     AH,2
        INT     1CH
LOOP:
        CMP     ES:STATUS,Ø             ;割込発生のチェック
        JE      LOOP
        MOV     AH,9                    ;割込発生メッセージを表示
        MOV     DX,OFFSET MSG_INT
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  タイマー処理ルーチン
INTR    SEGMENT
        ASSUME  CS:INTR
I_ENT:  MOV     CS:STATUS,ØFFH          ;STATUS=FFH(割込発生状態)
        IRET
;
STATUS  DB      Ø
INTR    ENDS
;#########  データ領域
DATA    SEGMENT
MSG_W   DB      'タイマー割込待ち．少し待って下さい．
        DB      ØDH,ØAH,'$'
MSG_INT DB      '5秒経過．タイマー割込発生．'
        DB      ØDH,ØAH,'$'
DATA    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      1ØØH DUP (?)
STACK   ENDS
        END     START
```

# 第2章

# キーボードBIOS

●キーボード BIOS 機能一覧

| AH レジスタ | 機能 | 備考 |
|---|---|---|
| 00H | キーデータの読み出し | |
| 01H | キーバッファ状態のセンス | |
| 02H | シフトキー状態のセンス | |
| 03H | キーボードインターフェイスの初期化 | |
| 04H | キー入力状態のセンス | |
| 05H | バッファからのキーコードの読み出し | 新規 |
| 06H | バッファの初期化 | 新規 |
| 07H | シフト状態とキーデータの読み出し | 新規 |
| 08H | シフト状態とキーデータのセンス | 新規 |
| 09H | キーデータの作成 | 新規 |

## 2.1 キーボード BIOS 共通情報

### (1) キーボード BIOS の主な特徴

・キーボードの生成するキーコードと，生成された時のシフト状態をバッファに格納する．

・要求により，バッファに格納したシフト状態とキーコードからキーデータを生成する．

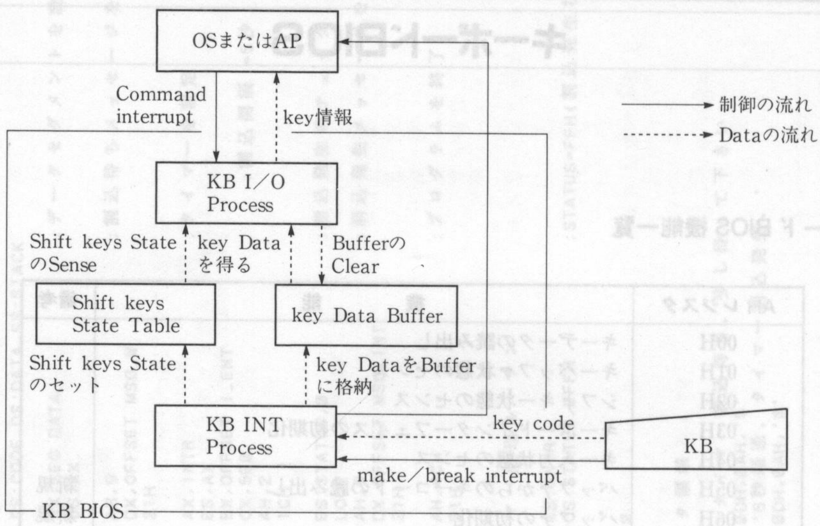
・ハードウェア割り込みにより内部割り込みを発生させる．

・与えられたキーコードとシフト状態をアスキーコードに変換する．

注：キーコード

キーが押された・離された時に，割り込みと同時にキーボードが生成するキーの値．

### (2) キーボード BIOS の構造




### (3) バッファオーバーフロー

バッファがオーバーフローした場合にはブザーを鳴す(約0.5秒)．

ただし，BIOS_FLAG のビット5がONの時はブザーを鳴さない．

### (4) ハードエラー

キーボードからデータを受け取る場合に次のエラーが考えられる．

① キーコード以外のデータを受け取った．

② Parity Error

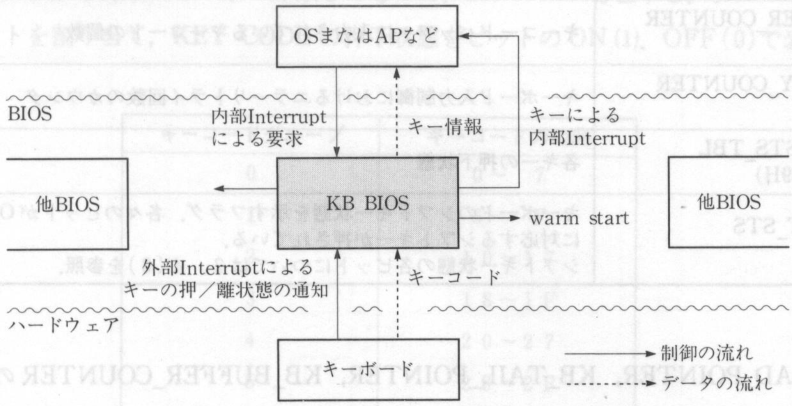
③ Overrun Error

④ Framing Error

①の時，キーデータとして扱う．

②～④の時，Error Retry を3回行う．4回めはキーコードとして扱う．

### (5) 本ルーチンの位置付け




### (6) 上位モジュール間インターフェイス

上位モジュールとのインターフェイスは次の2種類ある．

① キーボードI/O割り込みによるコマンドの実行

② キーによる内部割り込み

### (7) システム共通域

本ルーチンでは，ローカルRAMエリアとして次のものを使用する．

① バッファ管理／シフト状態管理関連

| システム共通域<br>シンボル名・絶対番地 | 用　　途 |
|---|---|
| KB_SHIFT_COD<br>(408H～40FH) | シフトキーとして扱うキーコード．バイトオフセットはKB_SHIFT_STSのビットオフセットに対応 |
| KB_BUFFER_ADR<br>(410H～413H) | バッファの開始アドレス |
| KB_ENTRY_TBL_ADR<br>(414H～417H) | ENTRY_TBLの開始アドレス |
| KB_INT_ADR<br>(418H～41BH) | 内部割り込みのためのテーブル(INT_TBL)開始アドレス |
| KB_KEY_BUFFER<br>(502H～521H) | キーボードからの入力キーコードを2バイト単位で最大16個格納するバッファ |
| KB_BUFFER_SIZ<br>(522H) | キーバッファサイズ |
| KB_FLG<br>(523H) | キーボードフラグ |
| KB_HEAD_POINTER<br>(524H，525H) | キーコードバッファ(KB_BUFFER)内の先頭のキーコードの位置を指すポインタ |
| KB_TAIL_POINTER<br>(526H，527H) | 既にキーコードバッファに格納されているエリアの最終アドレス＋1(未使用エリアの先頭．次の入力キーコードが格納されるアドレス)を指すポインタ |

| システム共通域<br>シンボル名・絶対番地 | 用　　途 |
|---|---|
| KB_BUFFER_COUNTER<br>(528H) | キーコードバッファに入力されているキーコードの個数 |
| KB_RETRY_COUNTER<br>(529H) | キーボード入力制御におけるエラーリトライ回数のカウンタ |
| KB_KEY_STS_TBL<br>(52AH～539H) | 各キーの押下状態 |
| KB_SHIFT_STS<br>(53AH) | キーボードのシフトキー状態を示すフラグ．各々のビットがON(1)の時に対応するシフトキーが押されている．<br>シフトキー状態の各ビットについては2．4(3)を参照． |

② KB_HEAD_POINTER, KB_TAIL_POINTER, KB_BUFFER_COUNTERの更新

1）バッファ内のキー情報の引き取り要求時

ⓐ KB_HEAD_POINTERの指すキーコードとシフト状態によってキーデータを生成する．

ⓑ KB_HEAD_POINTERを更新する．

ⓒ KB_BUFFER_COUNTERをデクリメントする．

2）キー入力時

ⓐ KB_TAIL_POINTERの指す所にキー情報を格納する．

ⓑ KB_TAIL_POINTERを更新する．

ⓒ KB_BUFFER_CONTERをインクリメントする．

③ キーの状態のセンス関連

KB_KEY_STS_TBL

・キーコードとビット位置を対応づける．

・同時に押下されているキーコードに対応するビットがONになる．

初期状態　all 0

更新　　bit ON ……メイク割り込み時

　　　　bit OFF……ブレイク割り込み時

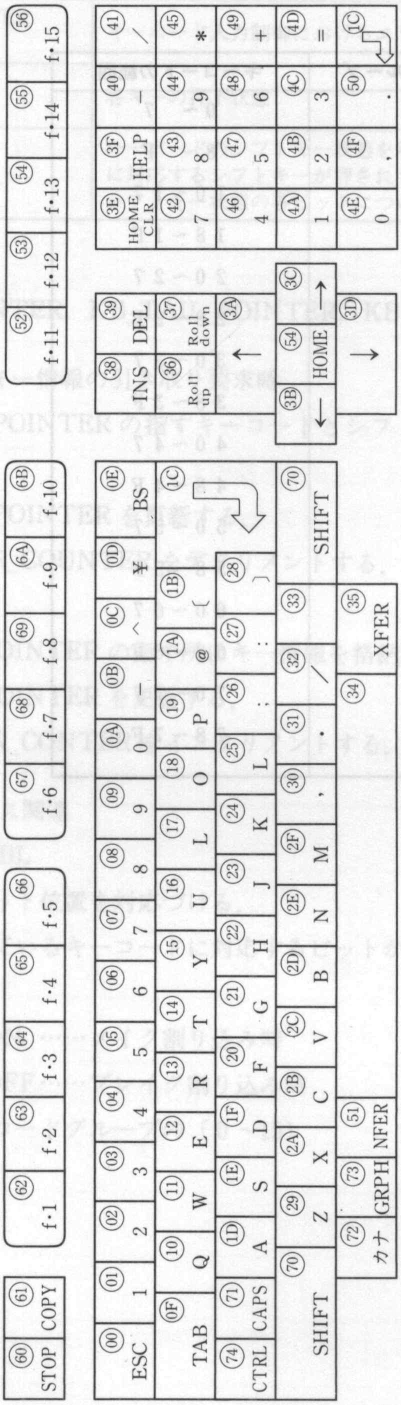
範囲　　キーコードグループの〔0～15〕

注：キーコードグループ

KEY CODE を昇順に8つずつ分類したもので，16グループ存在する．グループに対して1バイトを割り当て，KEY CODE の押下状態をビットの ON(1)．OFF(0)で表現する．

| キーコードグループ | キーコードの範囲 |
|---|---|
| 0 | 0～ 7 |
| 1 | 8～ F |
| 2 | 10～17 |
| 3 | 18～1F |
| 4 | 20～27 |
| 5 | 28～2F |
| 6 | 30～37 |
| 7 | 38～3F |
| 8 | 40～47 |
| 9 | 48～4F |
| A | 50～57 |
| B | 58～5F |
| C | 60～67 |
| D | 68～6F |
| E | 70～77 |
| F | 78～7F |

**〈キー配列とキーコードとの対応表〉**




注：数字はメイク割り込みコードを示す．ブレイク割り込みコードはメイクに80Hを加えた値となる．

## 〈キーコードとキーデータ対応表〉

### ● Key Code (00H～0FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | ESC | 00 | 1B | 00 | 1B | 00 | 1B | 00 | 1B | 00 | 1B | 00 | 1B | 00 | 1B | | |
| 01 | !<br>1 ヌ | 01 | 31 | 01 | 21 | 01 | 31 | 01 | 21 | 01 | C7 | 01 | C7 | | | | |
| 02 | 〃<br>2 フ | 02 | 32 | 02 | 22 | 02 | 32 | 02 | 22 | 02 | CC | 02 | CC | | | | |
| 03 | # ァ<br>3 ア | 03 | 33 | 03 | 23 | 03 | 33 | 03 | 23 | 03 | B1 | 03 | A7 | | | | |
| 04 | $ ゥ<br>4 ウ | 04 | 34 | 04 | 24 | 04 | 34 | 04 | 24 | 04 | B3 | 04 | A9 | | | | |
| 05 | % ェ<br>5 エ | 05 | 35 | 05 | 25 | 05 | 35 | 05 | 25 | 05 | B4 | 05 | AA | 05 | F2 | | |
| 06 | & ォ<br>6 オ | 06 | 36 | 06 | 26 | 06 | 36 | 06 | 26 | 06 | B5 | 06 | AB | 06 | F3 | | |
| 07 | ’ ャ<br>7 ヤ | 07 | 37 | 07 | 27 | 07 | 37 | 07 | 27 | 07 | D4 | 07 | AC | 07 | F4 | | |
| 08 | ( ュ<br>8 ユ | 08 | 38 | 08 | 28 | 08 | 38 | 08 | 28 | 08 | D5 | 08 | AD | 08 | F5 | | |
| 09 | ) ョ<br>9 ヨ | 09 | 39 | 09 | 29 | 09 | 39 | 09 | 29 | 09 | D6 | 09 | AE | 09 | F6 | | |
| 0A | ヲ<br>0 ワ | 0A | 30 | 0A | 30 | 0A | 30 | 0A | 30 | 0A | DC | 0A | A6 | 0A | F7 | | |
| 0B | =<br>－ ホ | 0B | 2D | 0B | 3D | 0B | 2D | 0B | 3D | 0B | CE | 0B | CE | 0B | 8C | | |
| 0C | ‘<br>∧ ヘ | 0C | 5E | 0C | 60 | 0C | 5E | 0C | 60 | 0C | CD | 0C | CD | 0C | 8B | 0C | 1E |
| 0D | ¦<br>¥ ― | 0D | 5C | 0D | 7C | 0D | 5C | 0D | 7C | 0D | B0 | 0D | B0 | 0D | F1 | 0D | 1C |
| 0E | BS | 0E | 08 | 0E | 08 | 0E | 08 | 0E | 08 | 0E | 08 | 0E | 08 | 0E | 08 | 0E | 08 |
| 0F | TAB | 0F | 09 | 0F | 09 | 0F | 09 | 0F | 09 | 0F | 09 | 0F | 09 | 0F | 19 | 0F | 09 |

● Key Code (10H〜1FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | Qタ | 10 | 71 | 10 | 51 | 10 | 51 | 10 | 71 | 10 | C0 | 10 | C0 | 10 | 9C | 10 | 11 |
| 11 | Wテ | 11 | 77 | 11 | 57 | 11 | 57 | 11 | 77 | 11 | C3 | 11 | C3 | 11 | 9D | 11 | 17 |
| 12 | ィ<br>Eイ | 12 | 65 | 12 | 45 | 12 | 45 | 12 | 65 | 12 | B2 | 12 | A8 | 12 | E4 | 12 | 05 |
| 13 | Rス | 13 | 72 | 13 | 52 | 13 | 52 | 13 | 72 | 13 | BD | 13 | BD | 13 | E5 | 13 | 12 |
| 14 | Tカ | 14 | 74 | 14 | 54 | 14 | 54 | 14 | 74 | 14 | B6 | 14 | B6 | 14 | EE | 14 | 14 |
| 15 | Yン | 15 | 79 | 15 | 59 | 15 | 59 | 15 | 79 | 15 | DD | 15 | DD | 15 | EF | 15 | 19 |
| 16 | Uナ | 16 | 75 | 16 | 55 | 16 | 55 | 16 | 75 | 16 | C5 | 16 | C5 | 16 | F0 | 16 | 15 |
| 17 | Iニ | 17 | 69 | 17 | 49 | 17 | 49 | 17 | 69 | 17 | C6 | 17 | C6 | 17 | E8 | 17 | 09 |
| 18 | Oラ | 18 | 6F | 18 | 4F | 18 | 4F | 18 | 6F | 18 | D7 | 18 | D7 | 18 | E9 | 18 | 0F |
| 19 | Pセ | 19 | 70 | 19 | 50 | 19 | 50 | 19 | 70 | 19 | BE | 19 | BE | 19 | 8D | 19 | 10 |
| 1A | ~<br>@“ | 1A | 40 | 1A | 7E | 1A | 40 | 1A | 7E | 1A | DE | 1A | DE | 1A | 8A | 1A | 00 |
| 1B | {「<br>[。 | 1B | 5B | 1B | 7B | 1B | 5B | 1B | 7B | 1B | DF | 1B | A2 | | | 1B | 1B |
| 1C | ⏎ | 1C | 0D | 1C | 0D | 1C | 0D | 1C | 0D | 1C | 0D | 1C | 0D | 1C | 0D | 1C | 0D |
| 1D | Aチ | 1D | 61 | 1D | 41 | 1D | 41 | 1D | 61 | 1D | C1 | 1D | C1 | 1D | 9E | 1D | 01 |
| 1E | Sト | 1E | 73 | 1E | 53 | 1E | 53 | 1E | 73 | 1E | C4 | 1E | C4 | 1E | 9F | 1E | 13 |
| 1F | Dシ | 1F | 64 | 1F | 44 | 1F | 44 | 1F | 64 | 1F | BC | 1F | BC | 1F | E6 | 1F | 04 |

● Key Code (20H～2FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | Fハ | 20 | 66 | 20 | 46 | 20 | 46 | 20 | 66 | 20 | CA | 20 | CA | 20 | E7 | 20 | 06 |
| 21 | Gキ | 21 | 67 | 21 | 47 | 21 | 47 | 21 | 67 | 21 | B7 | 21 | B7 | 21 | EC | 21 | 07 |
| 22 | Hク | 22 | 68 | 22 | 48 | 22 | 48 | 22 | 68 | 22 | B8 | 22 | B8 | 22 | ED | 22 | 08 |
| 23 | Jマ | 23 | 6A | 23 | 4A | 23 | 4A | 23 | 6A | 23 | CF | 23 | CF | 23 | EA | 23 | 0A |
| 24 | Kノ | 24 | 6B | 24 | 4B | 24 | 4B | 24 | 6B | 24 | C9 | 24 | C9 | 24 | EB | 24 | 0B |
| 25 | Lリ | 25 | 6C | 25 | 4C | 25 | 4C | 25 | 6C | 25 | D8 | 25 | D8 | 25 | 8E | 25 | 0C |
| 26 | ＋<br>；レ | 26 | 3B | 26 | 2B | 26 | 3B | 26 | 2B | 26 | DA | 26 | DA | 26 | 89 | | |
| 27 | ＊<br>：ケ | 27 | 3A | 27 | 2A | 27 | 3A | 27 | 2A | 27 | B9 | 27 | B9 | 27 | 94 | | |
| 28 | }」<br>コム | 28 | 5D | 28 | 7D | 28 | 5D | 28 | 7D | 28 | D1 | 28 | A3 | | | 28 | 1D |
| 29 | ッ<br>Zツ | 29 | 7A | 29 | 5A | 29 | 5A | 29 | 7A | 29 | C2 | 29 | AF | 29 | 80 | 29 | 1A |
| 2A | Xサ | 2A | 78 | 2A | 58 | 2A | 58 | 2A | 78 | 2A | BB | 2A | BB | 2A | 81 | 2A | 18 |
| 2B | Cソ | 2B | 63 | 2B | 43 | 2B | 43 | 2B | 63 | 2B | BF | 2B | BF | 2B | 82 | 2B | 03 |
| 2C | Vヒ | 2C | 76 | 2C | 56 | 2C | 56 | 2C | 76 | 2C | CB | 2C | CB | 2C | 83 | 2C | 16 |
| 2D | Bコ | 2D | 62 | 2D | 42 | 2D | 42 | 2D | 62 | 2D | BA | 2D | BA | 2D | 84 | 2D | 02 |
| 2E | Nミ | 2E | 6E | 2E | 4E | 2E | 4E | 2E | 6E | 2E | D0 | 2E | D0 | 2E | 85 | 2E | 05 |
| 2F | Mモ | 2F | 6D | 2F | 4D | 2F | 4D | 2F | 6D | 2F | D3 | 2F | D3 | 2F | 86 | 2F | 0D |

● Key Code (30H～3FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 30 | <，，ネ | 30 | 2C | 30 | 3C | 30 | 2C | 30 | 3C | 30 | C8 | 30 | A4 | 30 | 87 | | |
| 31 | >。・ル | 31 | 2E | 31 | 3E | 31 | 2E | 31 | 3E | 31 | D9 | 31 | A1 | 31 | 88 | | |
| 32 | ？・／メ | 32 | 2F | 32 | 3F | 32 | 2F | 32 | 3F | 32 | D2 | 32 | A5 | 32 | 97 | | |
| 33 | ＿＼□ | 33 | FC | 33 | 5F | 33 | FC | 33 | 5F | 33 | DB | 33 | DB | | | 33 | 1F |
| 34 | (SPACE) | 34 | 20 | 34 | 20 | 34 | 20 | 34 | 20 | 34 | 20 | 34 | 20 | 34 | 20 | 34 | 20 |
| 35 | XFER | 35 | 00 | A5 | 00 | 35 | 00 | A5 | 00 | 35 | 00 | A5 | 00 | 35 | 00 | B5 | 00 |
| 36 | ROLL UP | 36 | 00 | 36 | 00 | 36 | 00 | 36 | 00 | 36 | 00 | 36 | 00 | 36 | 00 | 36 | 00 |
| 37 | ROLL DOWN | 37 | 00 | 37 | 00 | 37 | 00 | 37 | 00 | 37 | 00 | 37 | 00 | 37 | 00 | 37 | 00 |
| 38 | INS | 38 | 00 | 38 | 00 | 38 | 00 | 38 | 00 | 38 | 00 | 38 | 00 | 38 | 00 | 38 | 00 |
| 39 | DEL | 39 | 00 | 39 | 00 | 39 | 00 | 39 | 00 | 39 | 00 | 39 | 00 | 39 | 00 | 39 | 00 |
| 3A | ↑ | 3A | 00 | 3A | 00 | 3A | 00 | 3A | 00 | 3A | 00 | 3A | 00 | 3A | 00 | 3A | 00 |
| 3B | ← | 3B | 00 | 3B | 00 | 3B | 00 | 3B | 00 | 3B | 00 | 3B | 00 | 3B | 00 | 3B | 00 |
| 3C | → | 3C | 00 | 3C | 00 | 3C | 00 | 3C | 00 | 3C | 00 | 3C | 00 | 3C | 00 | 3C | 00 |
| 3D | ↓ | 3D | 00 | 3D | 00 | 3D | 00 | 3D | 00 | 3D | 00 | 3D | 00 | 3D | 00 | 3D | 00 |
| 3E | CLR | 3E | 00 | 5E | 00 | 3E | 00 | 5E | 00 | 3E | 00 | 5E | 00 | | | | |
| 3F | HELP | 3F | 00 | 3F | 00 | 3F | 00 | 3F | 00 | 3F | 00 | 3F | 00 | 3F | 00 | 3F | 00 |

● Key Code (40H～4FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | − | 40 | 2D | 40 | 2D | 40 | 2D | 40 | 2D | 40 | 2D | 40 | 2D | 40 | 2D | 40 | 2D |
| 41 | / | 41 | 2F | 41 | 2F | 41 | 2F | 41 | 2F | 41 | 2F | 41 | 2F | 41 | 2F | 41 | 2F |
| 42 | 7 | 42 | 37 | 42 | 37 | 42 | 37 | 42 | 37 | 42 | 37 | 42 | 37 | 42 | 98 | 42 | 37 |
| 43 | 8 | 43 | 38 | 43 | 38 | 43 | 38 | 43 | 38 | 43 | 38 | 43 | 38 | 43 | 91 | 43 | 38 |
| 44 | 9 | 44 | 39 | 44 | 39 | 44 | 39 | 44 | 39 | 44 | 39 | 44 | 39 | 44 | 99 | 44 | 39 |
| 45 | * | 45 | 2A | 45 | 2A | 45 | 2A | 45 | 2A | 45 | 2A | 45 | 2A | 45 | 95 | 45 | 2A |
| 46 | 4 | 46 | 34 | 46 | 34 | 46 | 34 | 46 | 34 | 46 | 34 | 46 | 34 | 46 | E1 | 46 | 34 |
| 47 | 5 | 47 | 35 | 47 | 35 | 47 | 35 | 47 | 35 | 47 | 35 | 47 | 35 | 47 | E2 | 47 | 35 |
| 48 | 6 | 48 | 36 | 48 | 36 | 48 | 36 | 48 | 36 | 48 | 36 | 48 | 36 | 48 | E3 | 48 | 36 |
| 49 | + | 49 | 2B | 49 | 2B | 49 | 2B | 49 | 2B | 49 | 2B | 49 | 2B | 49 | E0 | 49 | 2B |
| 4A | 1 | 4A | 31 | 4A | 31 | 4A | 31 | 4A | 31 | 4A | 31 | 4A | 31 | 4A | 93 | 4A | 31 |
| 4B | 2 | 4B | 32 | 4B | 32 | 4B | 32 | 4B | 32 | 4B | 32 | 4B | 32 | 4B | 8F | 4B | 32 |
| 4C | 3 | 4C | 33 | 4C | 33 | 4C | 33 | 4C | 33 | 4C | 33 | 4C | 33 | 4C | 92 | 4C | 33 |
| 4D | = | 4D | 3D | 4D | 3D | 4D | 3D | 4D | 3D | 4D | 3D | 4D | 3D | 4D | 96 | 4D | 3D |
| 4E | 0 | 4E | 30 | 4E | 30 | 4E | 30 | 4E | 30 | 4E | 30 | 4E | 30 | 4E | 9A | 4E | 30 |
| 4F | , | 4F | 2C | 4F | 2C | 4F | 2C | 4F | 2C | 4F | 2C | 4F | 2C | 4F | 90 | 4F | 3C |

● Key Code (50H～5FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | · | 50 | 2E | 50 | 2E | 50 | 2E | 50 | 2E | 50 | 2E | 50 | 2E | 50 | 9B | 50 | 2E |
| 51 | NFER | 51 | 00 | A1 | 00 | 51 | 00 | A1 | 00 | 51 | 00 | A1 | 00 | 51 | 00 | B1 | 00 |
| 52 | F·11 | 52 | 00 | C2 | 00 | 52 | 00 | C2 | 00 | 52 | 00 | C2 | 00 | | | D2 | 00 |
| 53 | F·12 | 53 | 00 | C3 | 00 | 53 | 00 | C3 | 00 | 53 | 00 | C3 | 00 | | | D3 | 00 |
| 54 | F·13 | 54 | 00 | C4 | 00 | 54 | 00 | C4 | 00 | 54 | 00 | C4 | 00 | | | D4 | 00 |
| 55 | F·14 | 55 | 00 | C5 | 00 | 55 | 00 | C5 | 00 | 55 | 00 | C5 | 00 | | | D5 | 00 |
| 56 | F·15 | 56 | 00 | C6 | 00 | 56 | 00 | C6 | 00 | 56 | 00 | C6 | 00 | | | D6 | 00 |
| 57 | | | | | | | | | | | | | | | | | |
| 58 | | | | | | | | | | | | | | | | | |
| 59 | | | | | | | | | | | | | | | | | |
| 5A | | | | | | | | | | | | | | | | | |
| 5B | | | | | | | | | | | | | | | | | |
| 5C | | | | | | | | | | | | | | | | | |
| 5D | | | | | | | | | | | | | | | | | |
| 5E | HOME | AE | 00 | AE | 00 | AE | 00 | AE | 00 | AE | 00 | AE | 00 | | | | |
| 5F | | | | | | | | | | | | | | | | | |

● Key Code (60H～6FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 60 | STOP | 60 | 00 | 60 | 00 | 60 | 00 | 60 | 00 | 60 | 00 | 60 | 00 | | 60 | 00 |
| 61 | COPY | 61 | 00 | 61 | 00 | 61 | 00 | 61 | 00 | 61 | 00 | 61 | 00 | | 61 | 00 |
| 62 | f・1 | 62 | 00 | 82 | 00 | 62 | 00 | 82 | 00 | 62 | 00 | 82 | 00 | | 92 | 00 |
| 63 | f・2 | 63 | 00 | 83 | 00 | 63 | 00 | 83 | 00 | 63 | 00 | 83 | 00 | | 93 | 00 |
| 64 | f・3 | 64 | 00 | 84 | 00 | 64 | 00 | 84 | 00 | 64 | 00 | 84 | 00 | | 94 | 00 |
| 65 | f・4 | 65 | 00 | 85 | 00 | 65 | 00 | 85 | 00 | 65 | 00 | 85 | 00 | | 95 | 00 |
| 66 | f・5 | 66 | 00 | 86 | 00 | 66 | 00 | 86 | 00 | 66 | 00 | 86 | 00 | | 96 | 00 |
| 67 | f・6 | 67 | 00 | 87 | 00 | 67 | 00 | 87 | 00 | 67 | 00 | 87 | 00 | | 97 | 00 |
| 68 | f・7 | 68 | 00 | 88 | 00 | 68 | 00 | 88 | 00 | 68 | 00 | 88 | 00 | | 98 | 00 |
| 69 | f・8 | 69 | 00 | 89 | 00 | 69 | 00 | 89 | 00 | 69 | 00 | 89 | 00 | | 99 | 00 |
| 6A | f・9 | 6A | 00 | 8A | 00 | 6A | 00 | 8A | 00 | 6A | 00 | 8A | 00 | | 9A | 00 |
| 6B | f・10 | 6B | 00 | 8B | 00 | 6B | 00 | 8B | 00 | 6B | 00 | 8B | 00 | | 9B | 00 |

備考：STOP，COPY は，内部割り込みを発生する．

● Key Code (70H～7FH)

| Key Code | Shift State / Key | Base | | Shift | | Capital | | Capital Shift | | カナ | | カナ・Shift | | Graphic | Control | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 70 | SHIFT | 70 | 00 | 70 | 00 | 70 | 00 | 70 | 00 | 70 | 00 | 70 | 00 | | 70 | 00 |
| 71 | CAPS | 71 | 00 | 71 | 00 | 71 | 00 | 71 | 00 | 71 | 00 | 71 | 00 | | 71 | 00 |
| 72 | カナ | 72 | 00 | 72 | 00 | 72 | 00 | 72 | 00 | 72 | 00 | 72 | 00 | | 72 | 00 |
| 73 | GRPH | 73 | 00 | 73 | 00 | 73 | 00 | 73 | 00 | 73 | 00 | 73 | 00 | | 73 | 00 |
| 74 | CTRL | 74 | 00 | 74 | 00 | 74 | 00 | 74 | 00 | 74 | 00 | 74 | 00 | | 74 | 00 |

備考：SHIFT～CTRL は Shift 状態を設定する．また Shift Keys State を更新する．

## 2.1.1 キーの扱い方

### (1) シフトキー

シフトキーは上位が定義する.

キーボード BIOS は，KB_SHIFT_COD に格納されているキーコードをシフトキーとみなす. KB_SHIFT_COD に格納されているキーが押下された時, KB_SHIFT_STS の対応するビットを ON にする.

また，離された時は対応するビットを OFF にする.

### (2) 内部割り込み発生キー

内部割り込みを発生させるキーは上位が定義する.

キーボード BIOS は，KB_INT_ADR でアドレスされる INT_TBL に格納されているキーを内部割り込みを発生させるキーとみなす.

### (3) 一般のキー

シフトキーと内部割り込みキー以外はすべて一般のキーとみなす.

このキーは，離された時は KB_KEY_STS_TBL を更新するのみである.

また, 押下された時は, その時のシフト状態と発生したキーコードを KB_KEY_BUFFER に格納し，KB_KEY_STS_TBL を更新する.

KB_KEY_BUFFER への格納の仕方

| 上位　1byte | 下位　1byte |
|---|---|
| Key CODE | SHIFT |

### (4) シフトキーの取得

現在のシフト状態を得たい時は KB_SHIFT_STS をセンス(Sence)しなければならない.

キーコードに対するシフト状態は，KB_HEAD_POINTER が指すシフト状態をセンスする.

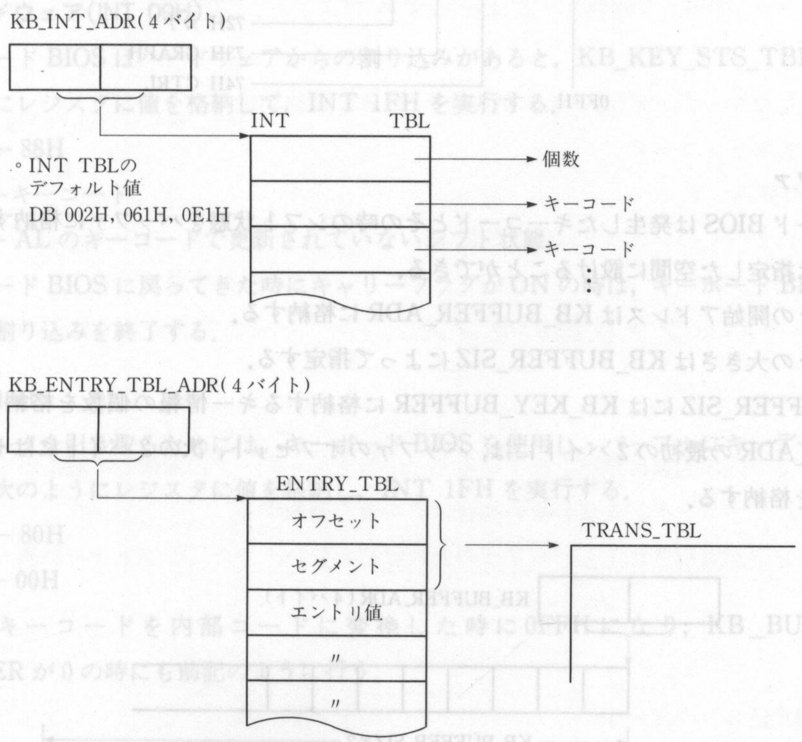
## 2.1.2　システム共通域

### (1) 内部割り込み

上位は内部割り込みを発生させるキーを任意に決められる.

BIOSが割り込み発生キーと他のキーと識別するために，INT_TBLが必要である．INT TBLはKB_INT_ADRによってアドレスされる．KB_INT_ADRの最初の2バイトにINT TBLのオフセット，次の2バイトにセグメントアドレスを格納する．INT_TBLの最初の1バイトにはINT_TBLに格納してあるキーコードの個数を格納する．

以後は割り込みの発生のキーコードを格納する．このキーコードの最上位ビットがON(1)の時INT 5を，OFFの時INT 6をそれぞれ発生させる．




ENTRY_TBLのエントリ値は，KB_SHIFT_STSによって参照する.

KB_SHIFT_STSの値が3ならば4番目のエントリ値をとり出す.

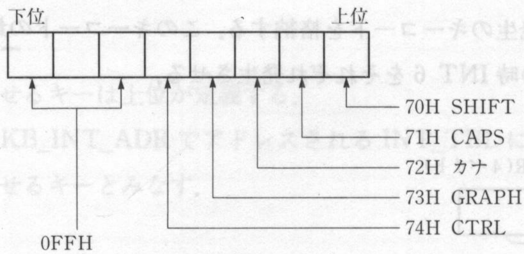
エントリ値は128バイト単位のTRNS_TBLの何番目のテーブルかを指す.

実際に取り出すアスキーコードは，エントリ値×128＋キーコードである.

(2) シフトキー

キーボード BIOS は他のキーとシフトキーを識別するために KB_SHIFT_COD を使用する.

キーコードが発生した時にキーボード BIOS は KB_SHIFT_COD と同じキーコードが格納されているかをみる. 格納されていれば, KB_SHIFT_STS を更新する. 同じキーコードが格納されていなければキーボード BIOS は発生したキーコードをシフトキーとみなさない.



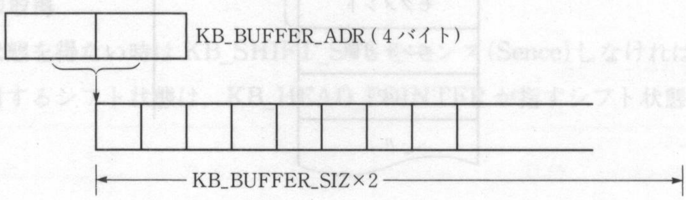

(3) バッファ

キーボード BIOS は発生したキーコードとその時のシフト状態をバッファに格納する. このバッファは指定した空間に設けることができる.

バッファの開始アドレスは KB_BUFFER_ADR に格納する.

バッファの大きさは KB_BUFFER_SIZ によって指定する.

KB_BUFFER_SIZ には KB_KEY_BUFFER に格納するキー情報の個数を格納する. KB BUFFER_ADR の最初の2バイトには, バッファのオフセット, 次の2バイトにはセグメントアドレスを格納する.




注 : バッファに関係のある共通域を変更した時は, バッファの初期化を実行しなければならない.

### (4) キーコード変換

キーボードBIOSはコード変換のためにTRANS_TBLにアスキーコードを持っている．シフト状態により，どのテーブルを参照するかを決めるためにENTRY_TBLを設ける．

ENTRY_TBLは，KB_ENTRY_TBL_ADRによってアドレスされる．

KB_ENTRY_TBL_ADRの最初の2バイトにはENTRY_TBLのオフセット，次の2バイトにはセグメントアドレスを格納する．

ENTRY_TBLの最初の2バイトにはTRANS_TBLのオフセット，次の2バイトにはセグメントアドレスを格納する．

## 2.1.3 割り込み処理

### (1) ハードウェア(INT 09H)

キーボードBIOSはハードウェアからの割り込みがあると，KB_KEY_STS_TBLを更新し，次のようにレジスタに値を格納して，INT 1FHを実行する．

AH ← 88H
AL ←キーコード

BL ← ALのキーコードで更新されていないシフト状態

キーボードBIOSに戻ってきた時にキャリーフラグがONの時は，キーボードBIOSは処理を中止し割り込みを終了する．

### (2) INT 18H

キーデータを引き取るためには，キーボードBIOSを使用し，バッファにキーデータがない場合は，次のようにレジスタに値を格納し，INT 1FHを実行する．

AH ← 80H
AL ← 00H

また，キーコードを内部コードに変換した時に0FFHになり，KB_BUFFER_COUNTERが0の時にも前記のように行う．

## 2.2 キーデータの読み出し

**(1) 機能**

キーバッファの先頭に格納されているキー情報をキーデータコードに変換して読み出す．キーバッファ内にキーコードが格納されていなければ，AX に 8000H を格納し，INT 1F を実行する．キーボード BIOS に制御が戻ってきた時にキーコードが格納されていなければ，格納されるまで待つ．

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 00H

**(3) 出力**

・読み出したキーデータ（2 バイトから成る）を AX に格納する．

AH ←スキャンコード

AL ←内部コード

AX 以外のすべてのレジスタは保証される．

## 2.3 キーバッファ状態のセンス

**(1) 機能**

キーバッファ内の先頭に格納されているキー情報をキーデータに変換して読み出す．もし，バッファ内にキーコードが存在しない場合はその旨を通知する．

このコマンドが実行されてもバッファの更新は行わない．また，シフトキー状態も変化しない．

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 01H

**(3) 出力**

・先頭に格納されているキーデータを AX に読み出す．

AH ←スキャンコード

AL ←内部コード

・AX に読み出したデータの状態

BH＝01H　ならば有効

BH＝00H　ならば無効

AX，BH 以外のすべてのレジスタは保証される．

### (4) 注意

本機能はセンス(sence)のみ行い，バッファの状態を変化させないので，通常は，このコマンドの後に「キーデータの読み出し」コマンドを実行させせることになる．

## 2.4　シフトキー状態のセンス

### (1) 機能

現在の押下されているシフトキー状態(KB_SHIFT_COD)を調べる．

シフトキーとは次のキーのことを指す．

SHIFT，CAPS，カナ，GRPH，CTRL．

これらのシフトキーの状態によって入力(押下)されたキーの対応する内部コードが決まる．

### (2) 入力

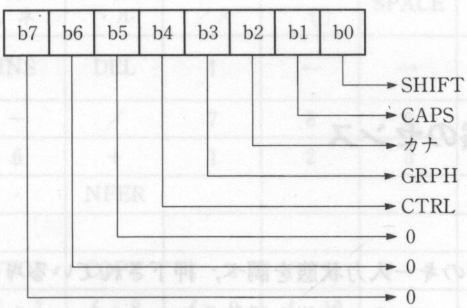
・内部割り込みコード← 18H

・AH ← 02H

### (3) 出力

・AL にシフトキー状態を格納する．

格納形式は次のとおりである．




当該シフトキーが押下されている状態の，対応ビットは1となる．

当該シフトキーが離されている状態のとき，対応ビットは0となる．

AX 以外のすべてのレジスタは保証される．

## 2.5 キーボードインターフェイスの初期化

### (1) 機能

KB BIOSが使用しているメモリエリアと，KBインターフェイスμPD8251Aを初期化する．新たに入力可能状態を作る．

### (2) 入力

・内部割り込みコード←18H

・AH←03H

### (3) 出力

AX以外のすべてのレジスタは保証される．

### (4) 処理

・KBインターフェイスμPD8251Aの初期化を行う．

・KB BIOSが使用している次のメモリエリアの初期化を行う．

KB_HEAD_POINTER

KB_TAIL_POINTER

KB_BUFFER_COUNTER

KB_RETRY_COUNTER

KB_KEY_STS_TBL

KB_SHIFT_STS

## 2.6 キー入力状態のセンス

### (1) 機能

当該キーコードグループのキー入力状態を調べ，押下されているキーは対応ビットを1，押下されていないキーは0として，その状態を通知する．

キーコードグループは，キーコード(00H～6BH)を，00H～07Hまでを0キーコードグループ，08H～0FHまでを1キーコードグループ，……のように区分し，16グループある．それぞれのグループは8個のキーから成っている．

### (2) 入力

・内部割り込みコード←18H

・AH←04H

・AL←キーコードグループ番号(00H～0FH)

**(3) 出力**

・AH←キーコードグループ内の8つのキー状態を8ビットで格納.

**(4) 処理**

該当するキー入力状態テーブル(KB_KEY_STS_TBL)を読み出し，AHに格納する.

**●キー入力状態テーブル**

| キーコードグループ＼ビット | $b_0$ | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ | $b_7$ |
|---|---|---|---|---|---|---|---|---|
| 0 | ESC | !<br>1ヌ | ”<br>2フ | #ァ<br>3ア | $ゥ<br>4ウ | %ェ<br>5エ | &ォ<br>6オ | ’ャ<br>7ヤ |
| 1 | (ュ<br>8ユ | )ョ<br>9ヨ | ヲ<br>0ワ | ニ<br>ーホ | ∧へ | ¦<br>¥ー | BS | TAB |
| 2 | Qタ | Wテ | ィ<br>ヨイ | Rス | Tカ | Yン | Uナ | Iニ |
| 3 | Oラ | Pセ | ~<br>@” | {「<br>〔。 | ↵ | Aチ | Sト | Dシ |
| 4 | Fハ | Gキ | Hワ | Jマ | Kノ | Lリ | +<br>;レ | *<br>:ケ |
| 5 | }」<br>コム | ッ<br>Zツ | Xサ | Cソ | Vヒ | Bコ | Nミ | Mモ |
| 6 | <,<br>，ネ | >。<br>・ル | ?・<br>／メ | ー<br>□ | SPACE | XFER | ROLL<br>UP | ROLL<br>DOWN |
| 7 | INS | DEL | ↑ | ← | → | ↓ | HOME<br>CLR | HELP |
| 8 | − | ／ | 7 | 8 | 9 | * | 4 | 5 |
| 9 | 6 | + | 1 | 2 | 3 | = | 0 | , |
| A | ・ | NFER | | | | | | |
| B | | | | | | | | |
| C | STOP | COPY | f・1 | f・2 | f・3 | f・4 | f・5 | f・6 |
| D | f・7 | f・8 | f・9 | f・10 | | | | |
| E | SHIFT | CAPS | カナ | GRPH | CTRL | | | |
| F | | | | | | | | |

## 2.7　バッファからのキーコードの読み出し

**(1) 機能**

バッファ内にキーコードが格納されているか否かを調べ，もしキーコードが存在したら，バッファの先頭に格納されているコードのキーデータを取得する．

**(2) 入力**

・内部割り込みコード←18H

・AH ← 05H

**(3) 出力**

・AH ←スキャンコード

・AL ←内部コード

BH＝01H　ならば有効

BH＝00H　ならば無効

AX，BH 以外のすべてのレジスタは保証される．

## 2.8　バッファの初期化

**(1) 機能**

バッファを初期化する．

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 06H

**(3) 出力**

なし．

AX 以外のすべてのレジスタが保証される．

## 2.9　シフトキー状態とキーデータの読み出し

### (1) 機能

キーバッファの先頭に格納されているキー情報のシフト状態と，生成したキーデータを読み出す．キーデータバッファ内にキーデータが格納されていなければ，AXに8000Hを格納し，INT 1FHを実行する．キーボードBIOSに制御が戻ってきた時にキーデータが格納されていなければ格納されるまで待つ．

### (2) 入力

・内部割り込みコード← 18H

・AH ← 07H

### (3) 出力

・AH ←スキャンコード

・AL ←内部コード

・BL ←シフト状態

AX，BL以外のすべてのレジスタは保証される．

## 2.10　シフトキー状態とキーデータのセンス

### (1) 機能

キーバッファの先頭に格納されているキー情報の，シフト状態と生成したキーデータを読み出す．

### (2) 入力

・内部割り込みコード← 18H

・AH ← 08H

### (3) 出力

・AH ←スキャンコード

・AL ←内部コード

・BL ←シフト状態

AX，BL以外のすべてのレジスタは保証される．

### (4) 注意

本機能はセンス(sence)のみを行い，バッファの状態を変化させない．通常はこのコマンドの後に「シフトキー状態とキーデータの読み出し」コマンドを実行させることになる．

## 2.11　キーデータの作成

**(1) 機能**

与えられたキーコードとシフト状態によりキーデータを生成する.

**(2) 入力**

・内部割り込みコード←18H

・AH←09H

・AL←キーコード

・BL←シフト状態

**(3) 出力**

・AH←スキャンコード

・AL←内部コード

AX, BL以外のすべてのレジスタは保証される.

● **サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである.
それ以外の環境での動作は保証できない.

```
;###########################################################
;#                                                         #
;#       キー情報を取り込む　（1）                          #
;#                                                         #
;###########################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,9                    ;入力を促す
        MOV     DX,OFFSET MSG
        INT     21H
;
        PUSH    DS
GET_KB:                                 ;リターンキー入力まで繰返し
        MOV     AH,Ø                     ;{キーデータを入力
        INT     18H
;
        MOV     AH,2                     ;入力データを表示
        MOV     DL,AL
        INT     21H
                                         ;}
        CMP     DL,ØDH
        JNE     GET_KB
        POP     DS
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#######   データ領域
DATA    SEGMENT
MSG     DB      ØDH,ØAH,'何かキーを押して下さい．'
        DB      'リターンキーを押すと終了します．',ØDH,ØAH,'$'
DATA    ENDS

;#######   スタック領域
STACK   SEGMENT STACK
        DW      4ØH DUP (?)
STACK   ENDS

        END     START
```

```
;###########################################################
;#                                                         #
;#       キー情報を取り込む　（2）                          #
;#                                                         #
;###########################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,9                    ;入力を促す
        MOV     DX,OFFSET MSG1
        INT     21H
KB:
        PUSH    DS
        MOV     AH,1                    ;キーセンス
        INT     18H
        CMP     BH,Ø
        JE      KB
        POP     DS
DISP:
        MOV     ASC,AL                  ;入力データを表示
        MOV     AH,9
        MOV     DX,OFFSET MSG2
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#######   データ領域
DATA    SEGMENT
MSG1    DB      1BH,'[2J','何かキーを押して下さい．',ØDH,ØAH,'$'
MSG2    DB      ØDH,ØAH,'KB_BUFFER_COUNTER が指すキーは，'
ASC     DB      ?
        DB      2ØH,'です．',ØDH,ØAH,ØAH,'$'
DATA    ENDS
;

;#######   スタック領域
STACK   SEGMENT STACK
        DW      4ØH DUP (?)
STACK   ENDS

        END     START
```

## 2.11 キーデータの作成

### (1) 機能

与えられたキーコードとシフト状態によりキーデータを生成する。

### (2) 入力

- 内部割り込みコード← 18H
- AH ← 09H
- AL ←キーコード
- BL ←シフト状態

### (3) 出力

- AH ←スキャンコード
- AL ←内部コード

AX，BL 以外のすべてのレジスタは保証される。

# 第3章

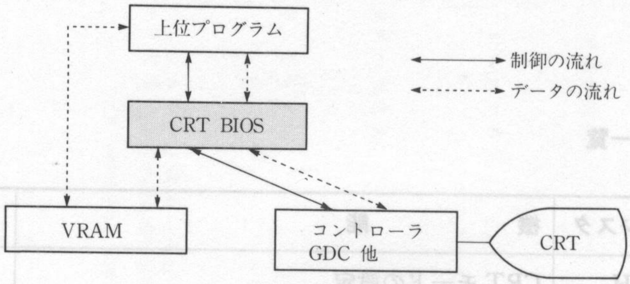
# CRT BIOS

● CRT BIOS 機能一覧

| AH レジスタ | 機　　能 | 備考 |
|---|---|---|
| 0AH | CRT モードの設定 | 変更 |
| 0BH | CRT モードのセンス | 変更 |
| 0CH | テキスト画面の表示開始 | |
| 0DH | テキスト画面の表示停止 | |
| 0EH | 1つの表示領域の設定 | |
| 0FH | 複数の表示領域の設定 | |
| 10H | カーソルタイプの設定 | |
| 11H | カーソルの表示 | |
| 12H | カーソル表示の停止 | |
| 13H | カーソル位置の設定 | |
| 14H | フォントパターンの読み出し | 無効 |
| 16H | テキスト VRAM の初期化 | |
| 1AH | ユーザー文字の定義 | 無効 |
| 1BH | KCG アクセスモードの設定 | |
| 1CH | CRT の初期化 | 新規 |
| 1DH | 表示幅の設定 | 新規 |
| 1EH | カーソルタイプの設定(XA モード) | 新規 |
| 1FH | フォントパターンの読み出し(24ドット) | 新規 |
| 20H | ユーザー文字の定義(24ドット) | 新規 |
| 21H | メモリスイッチの読み出し | 新規 |
| 22H | メモリスイッチの書き込み | 新規 |
| 19H | ライトペン押下状態の初期化 | 無効 |
| 15H | ライトペン位置の読み出し | 無効 |
| 17H | ブザーの起呼 | |
| 18H | ブザーの停止 | |
| 23H | ブザー周波数の設定 | 新規 |
| 24H | ブザーの時間設定と鳴動 | 新規 |

## 3.1　CRT BIOS 共通情報

### (1) データ関連図



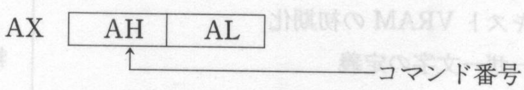

### (2) 上位プログラムとのインターフェイス

CRT BIOS は，上位プログラムからの内部割り込みによって起動される．上位プログラムは，INT 18H を実行することにより，CRT BIOS に制御を渡す．

①レジスタインターフェイス

a）コマンド

上位プログラムは，INT 18H 発行時，AH レジスタにコマンド番号を格納し，CRT BIOS の動作を決定する．



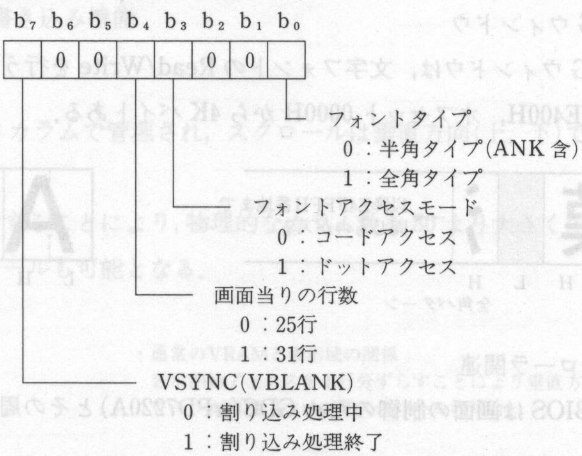

注：INT 18H の割り込みベクタは，キーボード BIOS と共有するため，AH レジスタに格納する値には注意すること．

b）パラメータ，データ

コマンド以外に，それぞれのコマンド動作に必要なパラメータ，データも上位プログラムよりレジスタで受け取る．

### (3) システム共通域

| システム共通域<br>シンボル名・絶対番地 | 用　途 |
|---|---|
| CAL_BEEP_TIME<br>(448H,449H) | ビープ機能のビープ継続時間(10msec) |
| CAL_TONE<br>(44AH, 44BH) | ビープの音調 |

| システム共通域<br>シンボル名・絶対番地 | 用　　　　　　途 |
|---|---|
| CRT_RASTER<br>(53BH) | CRTの行単位のラスタ数－1が格納される |
| CRT_CNT<br>(53DH) | 割り込み作業で使用する作業用カウンタ |
| CRT_PRM_OFST<br>(53EH, 53FH) | CRT制御用パラメータブロックのオフセット |
| CRT_PRM_SEG<br>(540H, 541H) | CRT制御用パラメータブロックのセグメント |
| CRTV_OFST<br>(542H, 543H) | CRTV割り込み時に割り込み処理ルーチンを切り換える時元の割り込みベクタを待避するアドレスのオフセット |
| CRTV_SEG<br>(544H, 545H) | CRTV割り込み時に割り込み処理ルーチンを切り換える時元の割り込みベクタを待避するアドレスのセグメント |
| CRT_FLAG<br>(546H) | CRT BIOSが使用するフラグ域<br><br> |
| CRT_W_NO<br>(547H) | GDCに設定する部分画面(スクロールエリア)の個数が格納される |
| CRT_W_ADR<br>(548H，549H) | 表示画面のVRAM上の開始アドレス(オフセット)が格納される |
| CRT_W_RASTER<br>(54AH,54BH) | 表示画面全体のラスタ数が格納される |
| CRT_FONT | 文字コード(2バイト) |
| CRT_P1 | GDCに設定するパラメータ(1バイト) |
| CRT_P2 | GDCに設定するパラメータ(1バイト) |
| CRT_P3 | GDCに設定するパラメータ(1バイト) |
| BEEP_TONE | ビープの音調(2バイト) |

## (4) ハードウェアインターフェイス

### ① VIDEO RAM 関連

a）VRAMの形式

・VRAMは，コードとアトリビュートに分けられる．

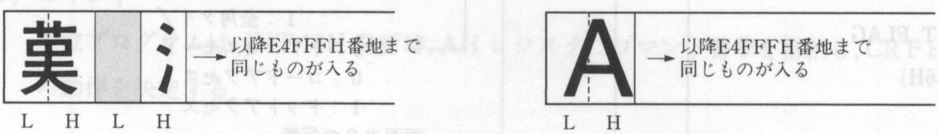
・コードメモリは4KB×2画面，アトリビュートメモリは2KB×2画面ある．

・VRAMは8086メモリ空間(E0000H～E3FDFH)に存在し，自由にRead/Writeができる．

| | | |
|---|---|---|
| E0000 | コード　H−1 | コード　L−1 |
| E1000 | コード　H−2 | コード　L−2 |
| E2000 | メモリは | アトリビュート　1 |
| E3000 | 存在しない | アトリビュート　2 |

b）CGウィンドウ

・CGウィンドウは，文字フォントのRead/Writeを行うためのメモリ空間で，セグメントE400H，オフセット0000Hから4Kバイトある．




全角パターン　　半角(ANK含)パターン

### ② コントローラ関連

CRT BIOSは画面の制御のためGDC($\mu$PD7220A)とその周辺LSIに対してIN/OUTを行う．

| I/Oアドレス | Read | Write |
|---|---|---|
| 60H | GDC STATUS | GDC PARAMETER |
| 62H | GDC DATA | GDC COMMAND |
| 64H | CRT TYPE | CRT INT RESET |
| 68H | ――― | Mode F/F |
| 70H | ――― | 52611　PL Reg. |
| 72H | ――― | 52611　BL Reg. |
| 74H | ――― | 52611　CL Reg. |
| 76H | ――― | 52611　SSL Reg.（注1） |
| 78H | ――― | 52611　SUR Reg.（注1） |
| 7AH | ――― | 52611　SDR Reg.（注1） |
| A1H | ――― | CG Write 2nd Byte Code |
| A3H | ――― | CG Write lst Byte Code |

注1：PC-98XL/XAはスムーススクロールの機能がないためアクセス不可

| $b_7$　　$b_0$<br>0000XXXY | Y＝0 | Y＝1 |
|---|---|---|
| XXX＝101 | KCG コードアクセスモード | ビットマップアクセス |
| XXX＝110 | 不揮発メモリ書き込み禁止 | 許可 |
| XXX＝111 | 表示停止 | 表示可 |

## 3.1.1　機　　能

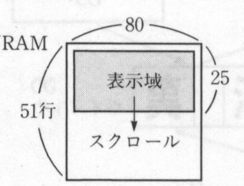
CRT BIOS は次の機能を有する．

(1) 画面設定機能
(2) 表示域設定機能
(3) カーソル制御機能
(4) フォントパターン読み出し／書込み機能
(5) ブザービープ機能
(6) 不揮発メモリ読み出し／書き込み機能

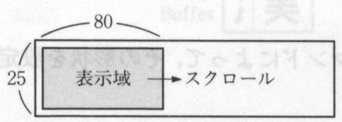
### (1) 画面設定機能

通常 VRAM は，物理的に 80 カラムで管理され，スクロールは垂直方向(上，下)で行われている．

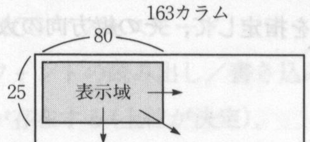
表示幅の設定コマンドを実行することにより，物理的なカラム数を 80 より大きくすることができるため，水平方向のスクロールも可能となる．




・通常のVRAMと表示域の関係
表示開始アドレスを1行分ずらすことにより垂直方向へのスクロールが可能（ページ間スクロールの場合）




・水平方向を最大にした場合．
表示開始アドレスを1カラム分ずらすことにより，水平方向のスクロールが可能



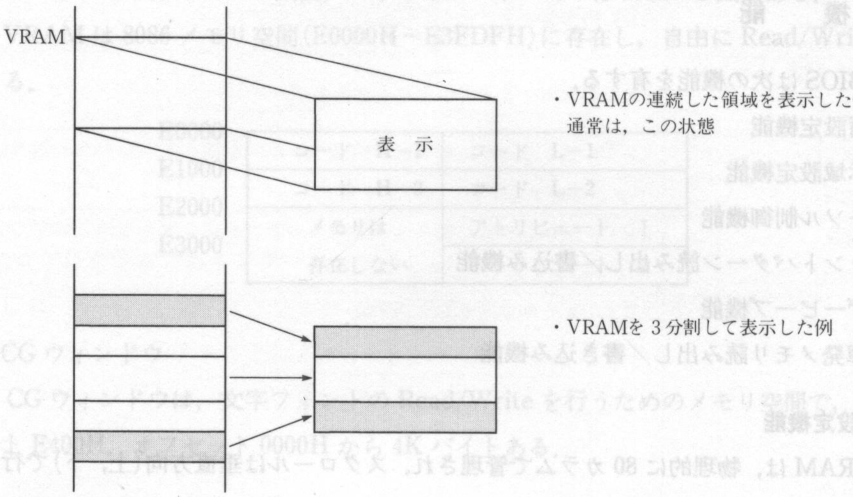

・水平方向，垂直方向に余裕をとった場合．
水平方向，垂直方向，斜めのスクロールが可能となる．

**(2) 表示域設定機能**

表示域設定機能では，画面分割，表示開始アドレスを制御する．

通常 VRAM の連続した領域が１画面として表示されるが，この機能を使用することにより，画面を最大４分割できる．

また，前述の画面設定機能と併用して画面のスクロールを行うこともできる．




**(3) カーソル制御機能**

カーソルに関する機能としては次のものがある．

・カーソル形状の設定

・ブリンクの有無指定

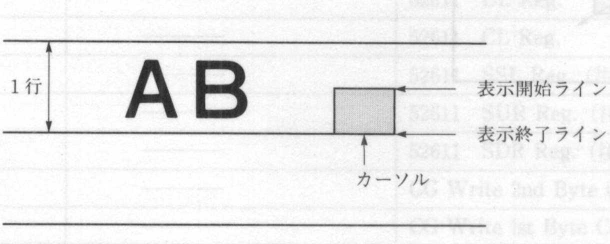
・カーソル表示の開始，停止

・カーソル表示アドレスの設定

① カーソル形状の設定

・カーソルは，カーソルタイプの設定（XA モード）コマンドによって，その形状を設定／変更することができる．

・カーソルの形状は，カーソルの表示開始，終了ラインを指定して，その縦方向の大きさを決定する（横方向の大きさは一定）．



② ブリンクの有無指定

カーソルタイプの設定(XA モード)コマンドでは，カーソルのブリンクの有無についても指定できる．

③ カーソル表示の開始，停止

カーソルの表示開始，停止は，それぞれカーソルの表示，カーソル表示の停止コマンドにて行う．

④ カーソル表示アドレスの設定

カーソル位置の設定コマンドで，VRAM アドレスを指定することにより，画面上の対応する位置に，カーソルを移動／表示させることができる．

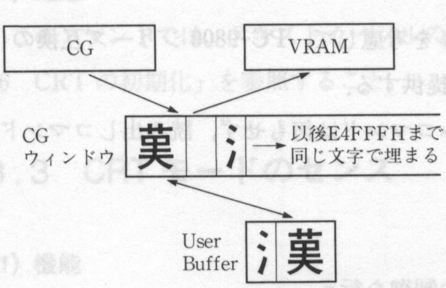
### (4) フォントパターン読み出し／書き込み機能

文字(ANK，漢字)のフォントパターンのCG からの読み出し，ユーザー定義文字(2 バイト系)パターンの，スタティック RAM への登録の機能もある．

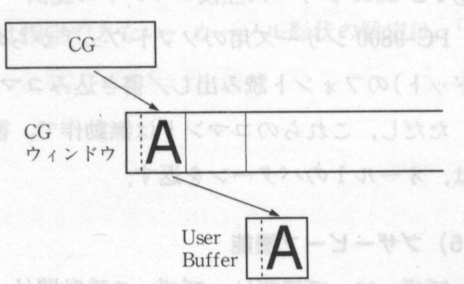
① フォント読み出し／書き込み動作

フォントの読み出し／書き込みは，CG ウィンドウを介して行う．CG ウィンドウは，0E4000H ~0E4FFFH のアドレス空間に存在する．

a）漢字フォントの読み出し，書き込み




b）ANK フォントの読み出し




② KCG アクセスモード

フォントの読み出し／書き込み時，テキスト画面表示中か否かによって2つのアクセスモードが存在する(上位が決定)．

a）テキスト画面に漢字を表示している場合，KCG アクセスモードは，コードアクセスになっていなければならない．

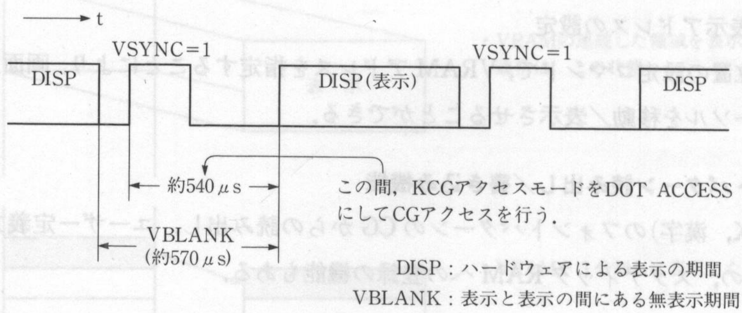
b）グラフィック画面のみの表示の場合，KCG アクセスモードは，ドットアクセスで良い．

③KCGアクセスモードのフォント読み出し／書き込み時のBIOSの動作

a）KCGアクセスモードがコードアクセスの場合

・KCGアクセスモードがコードアクセスの時，CRT BIOSは，画面の乱れ等を防ぐため CRTハードウェアのVSYNC割り込みを待つ．

・実際のCGアクセスは，VSYNCの立上がりから約540μSのVBLANK中に，KCGアクセスモードを一時的にドットアクセスにして行う．




b）KCGアクセスモードがドットアクセスの場合

・KCGアクセスモードがドットアクセスのとき，CRT BIOSは，VSYNC割り込みを待たずにCGアクセスを行う．

④PC-9800シリーズ互換コマンドの提供

PC-9800シリーズ用のソフトウェアからのコールを考慮して，PC-9800シリーズ互換の(16ドット)のフォント読み出し／書き込みコマンドも提供する．

ただし，これらのコマンドは無動作で，書き込みコマンドは何もせず，読み出しコマンドでは，オール1のパターンを返す．

### (5) ブザービープ機能

ブザービープ機能は，ブザーの鳴動開始，停止の制御を行う．

また，ブザーの周波数を変更する機能も提供する．

### (6) 不揮発メモリ読み出し／書き込み機能

この機能は不揮発メモリ(メモリスイッチ)の内容の読み込み／書き込みを行う．

不揮発メモリは通常書き込み禁止となっている．このためCRT BIOSは書き込み時に一時的に「書き込み許可」にする．

## 3.2　CRTモードの設定

**(1) 機能**

テキスト画面表示用GDCに対してCRT BIOSの初期化，モード設定を行う(PC-9800シリーズ互換).

**(2) 入力**

・内部割り込みコード←18H

・AH ← 0AH

・AL ←モード設定情報

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | 0 | 0 | 0 |

b3: KCGアクセス { 0：コードアクセス有効 / 1：ドットアクセス有効 }

b4: 画面当りの行数 { 0：25行 / 1：31行 }

**(3) 出力**

すべてのレジスタが保証される.

**(4) 処理**

このコマンドでは，テキストカーソルの形状等は指定できない．カーソル形状の設定は，「3.16　CRTの初期化」を参照すること.

## 3.3　CRTモードのセンス

**(1) 機能**

現在のKCGアクセスモードと，PC-9800シリーズ互換用の各フラグを返す.

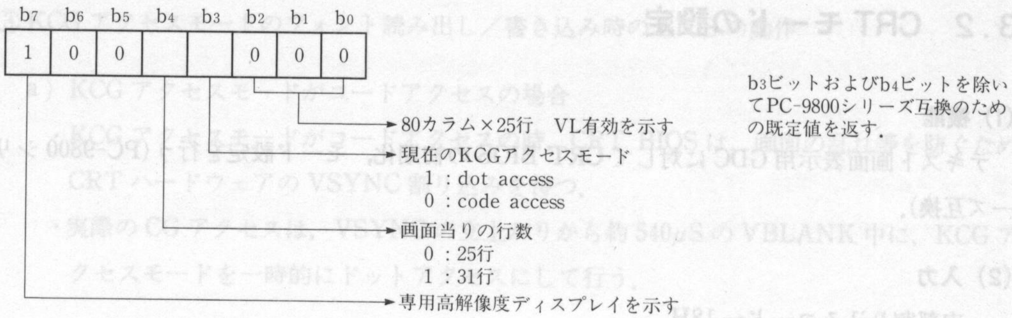
**(2) 入力**

・内部割り込みコード← 18H

・AH ← 0BH

**(3) 出力**

・各モードの状態が次の形式でALに格納される.




## 3.4　テキスト画面の表示開始

### (1) 機能

テキスト画面表示用GDCに表示開始要求を行う.

### (2) 入力

・内部割り込みコード←18H

・AH←0CH

### (3) 出力

すべてのレジスタが保証される.

## 3.5　テキスト画面の表示停止

### (1) 機能

テキスト画面表示用GDCに表示停止要求を行う.

### (2) 入力

・内部割り込みコード←18H

・AH←0DH

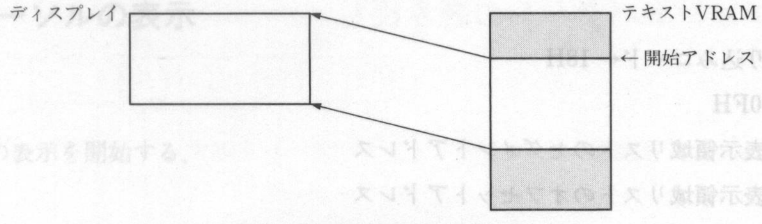
### (3) 出力

すべてのレジスタが保証される.

## 3.6　1つの表示領域の設定

### (1) 機能

テキスト画面1画面に表示する連続したVRAM領域を指定する. 領域の定義は, テキストVRAM上の開始アドレスを設定することによって行う.




(2) 入力

・内部割り込みコード←18H

・AH←0EH

・DX←表示する領域の開始アドレス．テキストVRAMのCPUアドレス20ビットのうち，下位16ビットを指定する(上位4ビットは0EHに固定されている)．

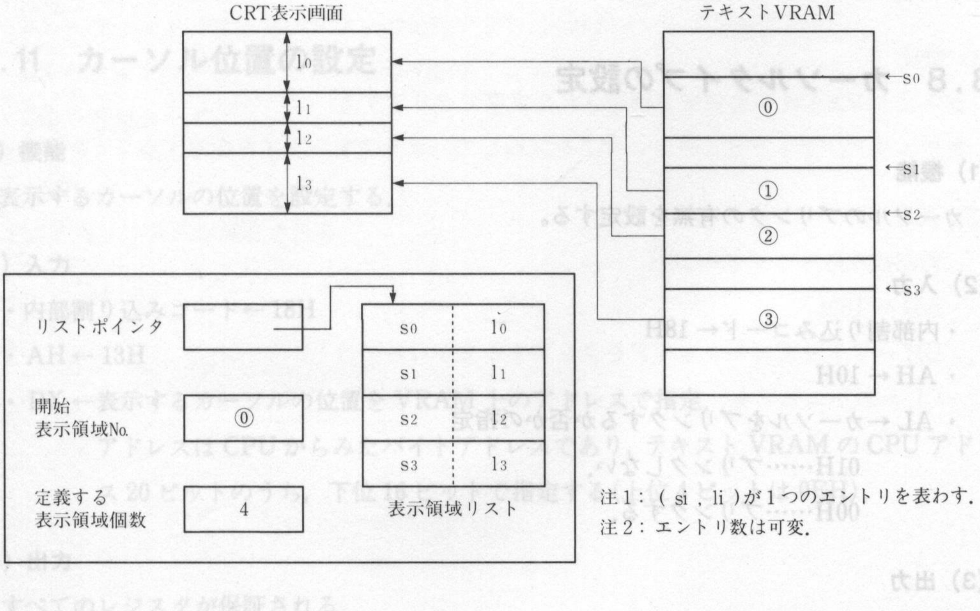
(3) 出力

すべてのレジスタが保証される．

## 3.7　複数個の表示領域の設定

(1) 機能

最大4個までの，テキスト画面へ表示する領域を設定するコマンドである．1つの表示領域の定義は，テキストVRAM上の開始アドレスとCRT上の表示行数によって定義する．4個までの表示領域を定義するために，メモリ上に次のような形式をしたリスト(表)を作成し，このアドレスを指定する．



**(2) 入力**

・内部割り込みコード← 18H

・AH ← 0FH

・BX ←表示領域リストのセグメントアドレス

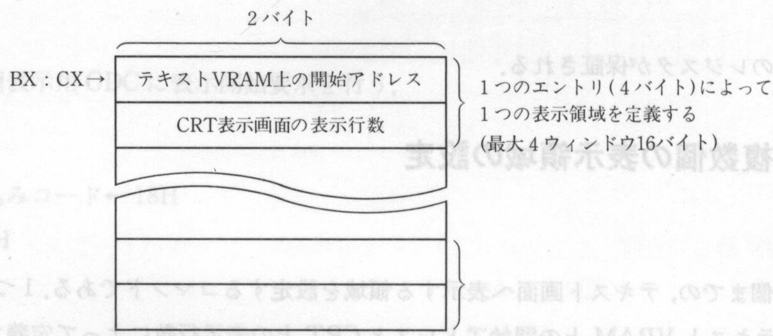
・CX ←表示領域リストのオフセットアドレス

・DH ←表示領域リストで最初に定義するエントリの表示領域番号(0～3)

(例えば，前回3分割しておき，今回2番目の表示領域を変更する場合，DH＝1とする．)

・DL ←表示領域リストで今回定義するエントリ個数(1～4)

・表示領域リストの形式




VRAMアドレスは，VRAMの先頭(E0000H)からのオフセットを格納する

**(3) 出力**

すべてのレジスタが保証される．

## 3.8 カーソルタイプの設定

**(1) 機能**

カーソルのブリンクの有無を設定する。

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 10H

・AL ←カーソルをブリンクするか否かの指定

01H……ブリンクしない．

00H……ブリンクする．

**(3) 出力**

すべてのレジスタが保証される．

## 3.9　カーソルの表示

(1) 機能

カーソルの表示を開始する.

(2) 入力

・内部割り込みコード← 18H

・AH ← 11H

(3) 出力

すべてのレジスタが保証される.

## 3.10　カーソル表示の停止

(1) 機能

カーソル表示を停止する.

(2) 入力

・内部割り込みコード← 18H

・AH ← 12H

(3) 出力

すべてのレジスタが保証される.

## 3.11　カーソル位置の設定

(1) 機能

表示するカーソルの位置を設定する.

(2) 入力

・内部割り込みコード← 18H

・AH ← 13H

・DX ←表示するカーソルの位置を VRAM 上のアドレスで指定.

アドレスはCPUからみたバイトアドレスであり，テキスト VRAM の CPU アドレス 20 ビットのうち，下位 16 ビットで指定する(上位 4 ビットは 0EH).

(3) 出力

すべてのレジスタが保証される.

## 3.12　フォントパターンの読み出し

### (1) 機能

PC-9800シリーズ互換用(オール1のフォントパターンを返す).

### (2) 入力

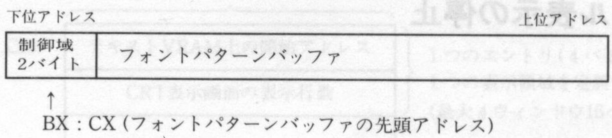
・内部割り込みコード← 18H

・AH ← 14H

・BX ←フォントパターンバッファの先頭アドレス(セグメントアドレス)

・CX ←フォントパターンバッファの先頭アドレス(オフセットアドレス)

バッファとして最低34バイト(32+2)を確保すること.




・DX ←展開するコードを設定

### (3) 出力

すべてのレジスタが保証される.

## 3.13　テキストVRAMの初期化

### (1) 機能

テキストVRAMの全領域を指定された文字でクリアする.

表示エリア，アトリビュートエリアを区別して，クリアする文字を指定することができる.

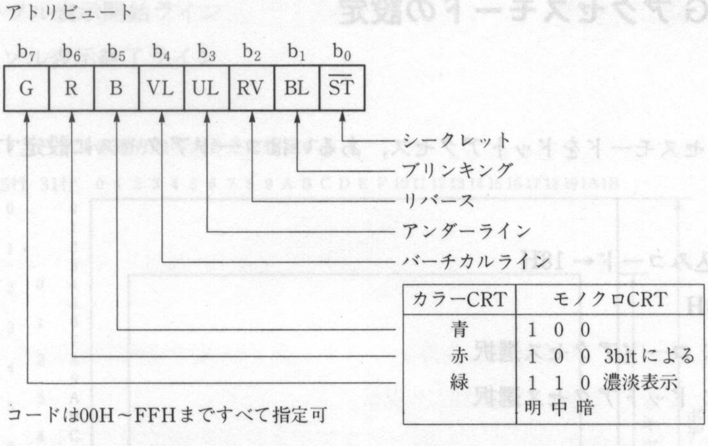
### (2) 入力

・内部割り込みコード← 18H

・AH ← 16H

・DH ←アトリビュートエリアをクリアする文字(1バイト)

・DL ←表示エリアをクリアする文字(1バイト)




### (3) 出力

すべてのレジスタが保証される.

## 3.14 ユーザー文字の定義

### (1) 機能

PC-9800シリーズ互換用(無動作).

### (2) 入力

・内部割り込みコード← 18H

・AH ← 1AH

・BX ←フォントパターンバッファの先頭アドレス(セグメントアドレス)

・CX ←フォントパターンバッファの先頭アドレス(オフセットアドレス)

・DX ←登録コード

### (3) 出力

すべてのレジスタが保証される.

## 3.15　KCG アクセスモードの設定

**(1) 機能**

KCG のアクセスモードをドットアクセス，あるいはコードアクセスに設定する．

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 1BH

・AL ← 0：コードアクセス選択

　　　1：ドットアクセス選択

**(3) 出力**

すべてのレジスタが保証される．

**(4) 処理**

このコマンド以後，システム共通域にモードを記憶し，以後のフォントの読み込み，フォントの書き出しコマンド時に参照する．

## 3.16　CRT の初期化

**(1) 機能**

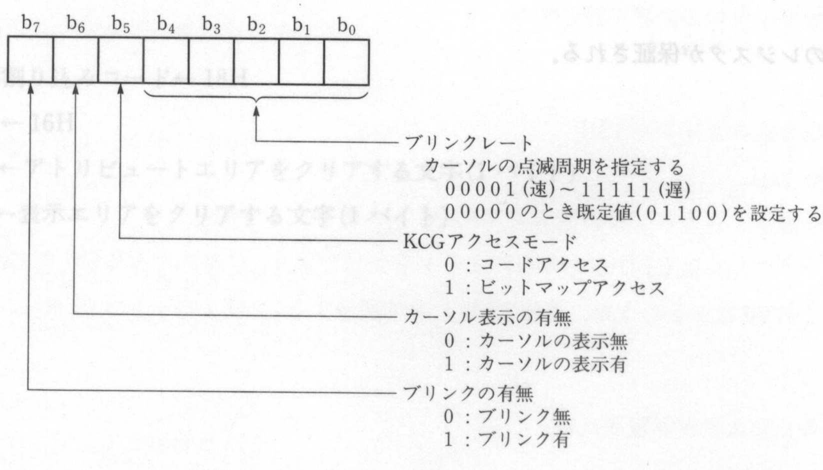
CRT BIOS の初期化を行い，カーソルの形状などを設定する

**(2) 入力**

・内部割り込み← 18H

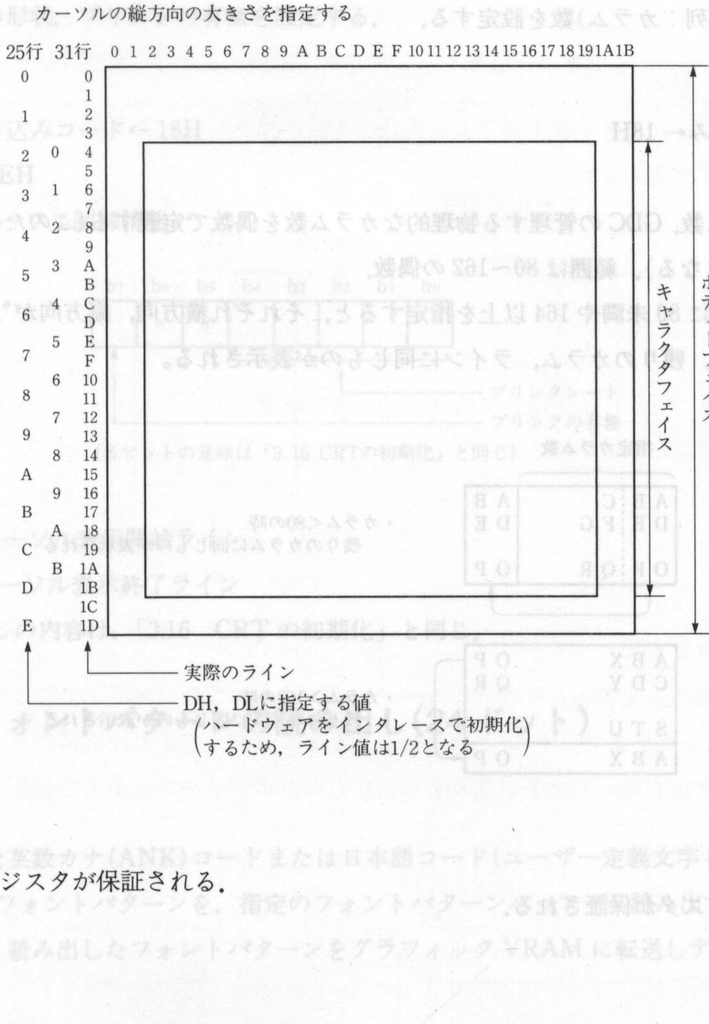
・AH ← 1CH

・AL ←カーソル制御情報



・DH ←カーソル表示開始ライン

・DL ←カーソル表示終了ライン




**(3) 出力**

すべてのレジスタが保証される．

## 3.17　表示幅の設定

### (1) 機能

VRAMの幅(列：カラム)数を設定する.

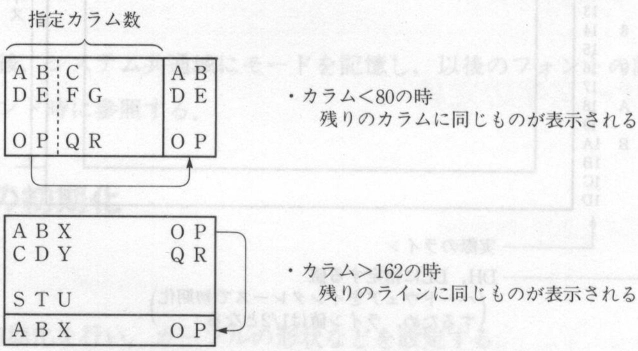
### (2) 入力

・内部割り込み←18H

・AH←1DH

・AL←カラム数. GDCの管理する物理的なカラム数を偶数で定義する(このため指定は2行づつになる). 範囲は80～162の偶数.

注意：カラム数に80未満や164以上を指定すると，それぞれ横方向，縦方向が"ラップアラウンド"し，残りのカラム，ラインに同じものが表示される。




### (3) 出力

すべてのレジスタが保証される.

## 3.18　カーソルタイプの設定(XA モード)

### (1) 機能

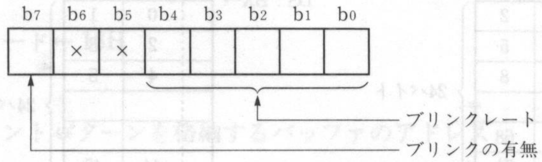
カーソルの形状，ブリンクの有無を設定する．

### (2) 入力

・内部割り込みコード← 18H

・AH ← 1EH

・AL ←カーソル制御情報




(各ビットの意味は「3.16 CRTの初期化」と同じ)

・DH ←カーソル表示開始ライン

・DL ←カーソル表示終了ライン

・DH，DL の内容は，「3.16　CRT の初期化」と同じ．

## 3.19　フォントパターンの読み出し(24 ドット)

### (1) 機能

指定された英数カナ(ANK)コードまたは日本語コード(ユーザー定義文字を含む)に対応する 24 ドットフォントパターンを，指定のフォントパターンバッファへ読み出す．

一般には，読み出したフォントパターンをグラフィック VRAM に転送しディスプレイに表示する．

### (2) 入力

・内部割り込みコード← 18H

・AH ← 1FH

・ DS/BX ←フォントパターンを格納するバッファのアドレス

・DX ← 展開するコードを設定(JIS コード／ASCII コード)

〈ANK コードの場合〉

・DH ← 00H

・DL ← ASCII コード

バッファサイズは 48 バイト(24×2)．

〈日本語コードの場合〉

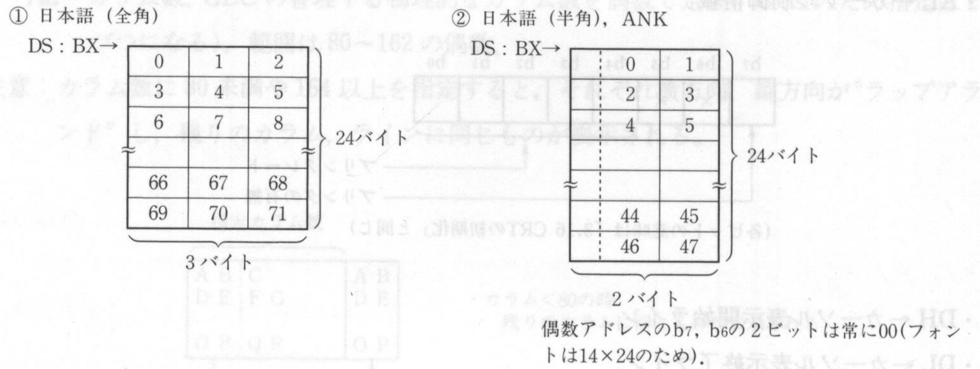
・DX ← JIS コード

バッファサイズは，全角の場合 72 バイト(24×3)，半角の場合，48 バイト(24×2).

**(3) 出力**

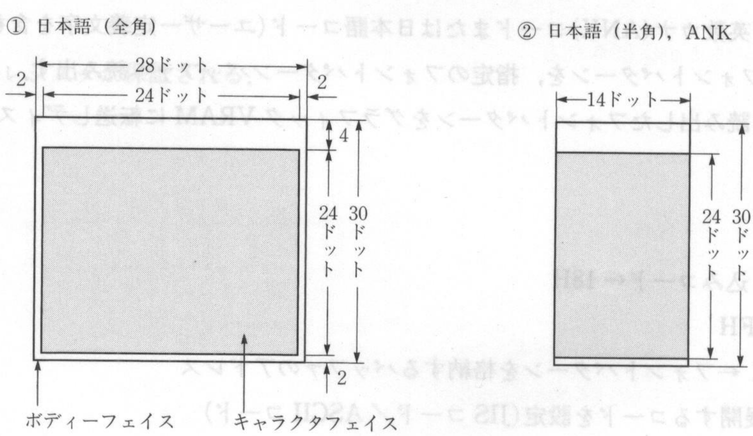
・出力バッファ形式

フォントパターンはキャラクタフェイス(注)で読み出される．各文字パターンのバッファへの格納形式は次のとおり．




注：キャラクタフェイス

画面表示の時の１文字の大きさ(ボディーフェイス)と，このコマンドで読み出される文字の大きさ(キャラクタフェイス)とは異なる．キャラクタフェイスは正味のフォントで，そのまわりに空白を置いて文字を見やすくしたものがボディーフェイスである．




**(4) 出力**

DL 以外のすべてのレジスタが保証される．

## 3.20　ユーザー文字の定義(24ドット)

### (1) 機能

ユーザーの作成した文字，記号の24ドットフォントパターンを，KCG RAMへ登録する．

このコマンド発行以降，テキスト画面，グラフィック画面への表示が可能になる(このコマンドはCRTに対してユーザー文字定義を行うもので，プリンタのユーザー文字定義とは別なので注意すること)．

ユーザーが定義可能な日本語コードは7621H～765FHである．

### (2) 入力

・内部割り込みコード←18H

・AH←20H

・DS/BX←フォントパターンを格納するバッファのアドレス

バッファの形式は「3.19　フォントパターンの読み出し(24ドット)」を参照のこと．

・DX←JIS日本語コード(7621H～765FH)

### (3) 出力

DL以外のすべてのレジスタが保証される．

## 3.21　メモリスイッチの読み出し

### (1) 機能

メモリスイッチの内容を読み出す．

### (2) 入力

・内部割り込みコード←18H

・AH←21H

・AL←スイッチ番号n(n=1～8)

### (3) 出力

・DL←メモリスイッチnの内容

DL以外のすべてのレジスタが保証される．

## 3.22　メモリスイッチの書き込み

### (1) 機能

メモリスイッチに内容を書き込む.

### (2) 入力

・内部割り込みコード← 18H

・AH ← 22H

・AL ←スイッチ番号 n(n=1〜8)

・DL ←メモリスイッチ n の内容

### (3) 出力

すべてのレジスタが保証される.

## 3.23　ライトペン押下状態の初期化

### (1) 機能

PC-9800 シリーズ互換用(無動作).

### (2) 入力

・内部割り込みコード← 18H

・AH ← 19H

### (3) 出力

AH 以外のすべてのレジスタが保証される.

## 3.24　ライトペン位置の読み出し

### (1) 機能

PC-9800 シリーズ互換用(既定値を返す).

### (2) 入力

・内部割り込みコード← 18H

・AH ← 15H

### (3) 出力

・AH ←ライトペンの状態(常に 01H).

・DX ←値はセットされない.

AH 以外のすべてのレジスタが保証される.

## 3.25　ブザーの起呼

**(1) 機能**

ブザーの鳴動を開始する.

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 17H

**(3) 出力**

すべてのレジスタが保証される.

## 3.26　ブザーの停止

**(1) 機能**

鳴動中のブザーを停止させる.

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 18H

**(3) 出力**

すべてのレジスタが保証される.

## 3.27　ブザー周波数の設定

**(1) 機能**

ブザーの周波数を設定する.

**(2) 入力**

・内部割り込みコード← 18H

・AH ← 23H

・DX ←音調(周波数)　範囲は 20H～8000H(Hz)

**(3) 出力**

すべてのレジスタが保証される.

## 3.28　ブザーの時間設定と鳴動

### (1) 機能

鳴動時間を設定して鳴動を行う．

### (2) 入力

・内部割り込みコード←18H

・AH←24H

・CX←鳴動時間

CX＝0H　ビープなし

CX＝1H～FFFFH(10msec～655350msec)

・DX←音調(周波数)　範囲は20H～8000H(Hz)

### (3) 出力

すべてのレジスタが保証される．

**● サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;####################################################################
;#                                                                  #
;#      C R T   B I O S  を 初 期 化 す る                          #
;#                                                                  #
;####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
START:
        MOV     AH,ØAH                  ;CRT-BIOSを初期化
        MOV     AL,ØØH                  ;アクセスモード
;                                       =CODE ACCESS/1画面25行
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;######### スタック領域
STACK   SEGMENT STACK
        DW      4ØH DUP (?)
STACK   ENDS
        END     START
```

```
;####################################################################
;#                                                                  #
;#      C R T  ステータスを読む                                     #
;#                                                                  #
;####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
        MOV     AH,ØBH                  ;CRT ステータスを取得
        INT     18H                     ;    al<--CRT ステータス
;
        MOV     CL,7H                   ;ステータスバイトを走査
        MOV     CH,AL
LOOP:
        MOV     SI,CX
        AND     SI,ØFFH
        SHL     SI,1
        SHL     SI,1
        MOV     AH,CH
        AND     AH,Ø1H
        JZ      SI_Ø
        INC     SI
        INC     SI
SI_Ø:
        MOV     DX,OFFSET MSGADDR[SI]   ;メッセージを選択
        MOV     AH,Ø9H                  ;メッセージを表示
        INT     21H
        SHR     CH,1
        DEC     CL
        JNS     LOOP
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;######### データ領域
DATA    SEGMENT
;                                 メッセージテーブル
MSGADDR DW      mØØ,mØ1,m1Ø,m11,m2Ø,m21,m3Ø,m31,m4Ø
        DW      m41,m5Ø,m51,m6Ø,m61,m7Ø,m71
MØØ     DB      '標準ディスプレイ',Ødh,Øah,'$'
MØ1     DB      '専用高解像度ディスプレイ',Ødh,Øah,'$'
M1Ø     DB      '$'
M11     DB      '$'
M2Ø     DB      '$'
M21     DB      '$'
M3Ø     DB      '２５行表示',Ødh,Øah,'$'
M31     DB      '３１行表示',Ødh,Øah,'$'
M4Ø     DB      'コードアクセス',Ødh,Øah,'$'
M41     DB      'ドットアクセス',Ødh,Øah,'$'
M5Ø     DB      'バーティカルライン',Ødh,Øah,'$'
```

```
M51      DB       '簡易グラフ',0dh,0ah,'$'
M60      DB       '行当たり８０字',0dh,0ah,'$'
M61      DB       '行当たり４０字',0dh,0ah,'$'
M70      DB       '$'
M71      DB       '$'
DATA     ENDS

;#########  スタック領域
STACK    SEGMENT STACK
         DW       80H DUP (?)
STACK    ENDS

         END      START
```

```
;##################################################################
;#                                                                #
;#   画面表示を開始する                                            #
;#                                                                #
;##################################################################
CODE     SEGMENT
         ASSUME   CS:CODE,SS:STACK
START:
         MOV      AH,0CH                        ;GDC にSTART コマンドを送信
         INT      18H
EXIT:
         MOV      AH,4CH                        ;プログラムを終了
         INT      21H
CODE     ENDS

;#########  スタック領域
STACK    SEGMENT STACK
         DW       80H DUP (?)
STACK    ENDS

         END      START
```

```
;##################################################################
;#                                                                #
;#       画面表示を停止／再開する                                  #
;#                                                                #
;##################################################################
CODE     SEGMENT
         ASSUME   CS:CODE,DS:DATA,SS:STACK
START:
         MOV      AX,SEG DATA                   ;データセグメントを設定
         MOV      DS,AX
;
         MOV      DX,OFFSET MSG                 ;メッセージを表示
         MOV      AH,09H
         INT      21H
         MOV      AH,01H                        ;キー入力待ち
         INT      21H
;
         MOV      AH,0DH                        ;GDC に STOP コマンドを送信
         INT      18H
;
         MOV      AH,01H                        ;キー入力待ち
         INT      21H
;
         MOV      AH,0CH                        ;GDC にSTART コマンドを送信
         INT      18H
EXIT:
         MOV      AH,4CH                        ;プログラムを終了
         INT      21H
CODE     ENDS

;#########  データ領域
DATA     SEGMENT PARA PUBLIC 'GRAPH'
MSG      DB       '何かキーを押すと，画面が消えますが，'
         DB       0DH,0AH,
         DB       'もう一度キーを押すと，もとに戻ります．'
         DB       0DH,0AH,'$'
DATA     ENDS

;#########  スタック領域
STACK    SEGMENT STACK
         DW       80H DUP (?)
STACK    ENDS

         END      START
```

```
;#####################################################################
;#                                                                   #
;#      ウィンドウを設定する                                         #
;#                                                                   #
;#####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA                   ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,ØEH                        ;表示領域を設定
        MOV     DX,ØØ4ØH                      ;  (VRAMのEØØØ:ØØ4Øから)
        INT     18H
;
        MOV     AH,Ø9H                        ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
        MOV     AH,Ø1H                        ;キー入力待ち
        INT     21H
;
        MOV     AH,ØEH                        ;表示領域をもとに戻す
        MOV     DX,ØH
        INT     18H
EXIT:
        MOV     AH,4CH                        ;プログラムを終了
        INT     21H
CODE    ENDS
;########  データ領域
DATA    SEGMENT
MSG     DB      '何かキーを押すと,標準の状態になります.'
        DB      Ødh,Øah,'$'
DATA    ENDS
;########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;#####################################################################
;#                                                                   #
;#      マルチウィンドウを設定する                                   #
;#                                                                   #
;#####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA                   ;データセグメントを設定
        MOV     DS,AX
;
        MOV     BX,AX                         ;4つのウィンドウを設定
        MOV     AH,ØFH
        MOV     CX,OFFSET PARAM                   ;パラメータオフセット
        MOV     DH,Ø                              ;開始ウィンドウの番号
        MOV     DL,4                              ;ウィンドウの数: 4
        INT     18H
;
        MOV     AH,Ø9H                        ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H

        MOV     AH,Ø1H                        ;キー入力待ち
        INT     21H
;
        MOV     AH,ØEH                        ;表示をもとにもどす
        MOV     DX,ØH
        INT     18H
EXIT:
        MOV     AH,4CH                        ;プログラムを終了
        INT     21H
CODE    ENDS
;########  データ領域
DATA    SEGMENT
PARAM   DW      2Ø,3,54Ø,4,1Ø6Ø,5,158Ø,6
MSG     DB      '何かキーを押すと, 標準の状態にもどります.'
        DB      ØDH,ØAH,'$'
DATA    ENDS
;########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;###############################################################
;#                                                             #
;#      カーソルタイプを設定する                               #
;#                                                             #
;###############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;                                       ;カーソルタイプを設定
        MOV     AH,10H
        MOV     AL,01H                      ;ブリンク"なし"
        INT     18H
;
        MOV     AH,11H                  ;カーソルを表示
        INT     18H
;
        MOV     AH,09H                  ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
;
        MOV     AH,01H                  ;キー入力待ち
        INT     21H
;
;                                       ;カーソルタイプを設定
        MOV     AH,10H
        MOV     AL,0H                   ;ブリンク"あり"
        INT     18H
;
        MOV     AH,11H                  ;カーソルを表示
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  データ領域
DATA    SEGMENT
MSG     DB      '何かキーを押すと,カーソルブリンクの状態になります.'
        DB      0DH,0AH,'$'
DATA    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS
        END     START
```

```
;###############################################################
;#                                                             #
;#      カーソルを表示する                                     #
;#                                                             #
;###############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
START:
        MOV     AH,11H                  ;カーソルを表示
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS
        END     START
```

```
;###############################################################
;#                                                             #
;#      カーソルの表示を停止する                               #
;#                                                             #
;###############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
START:
        MOV     AH,12H                  ;カーソルの表示を停止
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS
        END     START
```

```
;##################################################################
;#                                                                #
;#     指定した位置にテキストカーソルを表示する                   #
;#                                                                #
;##################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA                 ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,09H                      ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
;
        MOV     AH,13H                      ;カーソルをE000:200Hに表示
        MOV     DX,200
        INT     18H
;
        MOV     AH,01H                      ;キー入力待ち
        INT     21H
EXIT:
        MOV     AH,4CH                      ;プログラムを終了
        INT     21H
CODE    ENDS

;########  データ領域
DATA    SEGMENT
MSG     DB      '何かキーを押すと,カーソルが標準の位置にもどります.'
        DB      0DH,0AH,'$'
DATA    ENDS

;########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                                #
;#     フォントパターンを読み出す(PC-9800シリーズ互換)            #
;#                                                                #
;#          :"漢"のフォントを表示する                             #
;#                                                                #
;##################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA                 ;データセグメントを設定
        MOV     DS,AX
TOP:
        MOV     AH,09H                      ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
;
        MOV     BX,SEG BUFF_G               ;"漢"のフォントを読み込む
        MOV     CX,OFFSET BUFF_G
        MOV     DX,3441H
        MOV     AH,14H
        INT     18H
                                            ;フォントを表示する
        XOR     SI,SI
;                                             縦16行分繰返し
OL:     MOV     BH,BYTE PTR BUFF_G[SI]+2    ;{
        MOV     BL,BYTE PTR BUFF_G[SI]+3    ;{
;                                               横16ドット分繰返し
        MOV     CH,16                       ;{
;                                                 1ドット分表示
IL:     TEST    BX,8000H
        JS      O
X:      MOV     DX,OFFSET DOT_OFF                   ;bit=0なら'  '
        JMP     PUT
O:      MOV     DX,OFFSET DOT_ON                    ;bit=1なら'■'
PUT:    PUSH    CX                                  ;1文字表示
        MOV     AH,9
        INT     21H
        POP     CX
        SHL     BX,1                        ;}
        DEC     CH
        JNZ     IL
        MOV     AH,09H                      ;改行
        MOV     DX,OFFSET CRLF
        INT     21H
        INC     SI
        INC     SI
                                            ;}
        CMP     SI,32
        JNE     OL
EXIT:
        MOV     AH,4CH                      ;プログラムを終了
        INT     21H
CODE    ENDS
```

```
;#########  データ領域
DATA    SEGMENT
MSG     DB      'PC-9800シリーズとの互換用コマンドを使って'
        DB      0DH,0AH
        DB      '漢字"漢"のフォントパターンを示す',0DH,0AH,'$'
CRLF    DB      0DH,0AH,'$'
DOT_ON  DB      '■$'
DOT_OFF DB      ' $'
BUFF_G  DB      2+32 DUP (?)
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;#####################################################################
;#                                                                   #
;#      ライトペンポジションを得る（PC-9800シリーズ互換）            #
;#                                                                   #
;#####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,15H                  ;ポジションを得る
        INT     18H                     ;     (ステータスは常に01H)
;
        ADD     AH,30H                  ;ステータスをASCII文字に変換
        MOV     STATUS,AH
        MOV     AH,9                    ;ステータスを表示
        MOV     DX,OFFSET MSG
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT
MSG     DB      0DH,0AH,'ステータスに［'
STATUS  DB      ?
MES1    DB      '］が返されました.',0DH,0AH,'$'
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;###################################################################
;#                                                                 #
;#      テキスト VRAM を初期化する                                 #
;#                                                                 #
;###################################################################
CODE     SEGMENT
         ASSUME  CS:CODE,SS:STACK
START:
;                                         ;VRAM を初期化
         MOV     DH,81H                      ;アトリビュート：緑
         MOV     DL,'a'                      ;埋め字："a"
         MOV     AH,16H
         INT     18H
EXIT:
         MOV     AH,4CH                   ;プログラムを終了
         INT     21H
CODE     ENDS
;######### スタック領域
STACK    SEGMENT STACK
         DW      80H DUP (?)
STACK    ENDS
         END     START
```

```
;###################################################################
;#                                                                 #
;#      ブザー鳴動を開始・終了する                                 #
;#                                                                 #
;###################################################################
CODE     SEGMENT
         ASSUME  CS:CODE,DS:DATA,SS:STACK
START:   MOV     AX,SEG DATA              ;データセグメントを設定
         MOV     DS,AX
;
         MOV     AH,17H                   ;ブザーを鳴らす
         INT     18H
;
         MOV     AH,9                     ;メッセージを表示
         MOV     DX,OFFSET MSG
         INT     21H
;
         MOV     AH,1                     ;キー入力待ち
         INT     21H
;
         MOV     AH,18H                   ;ブザーを止める
         INT     18H
EXIT:
         MOV     AH,4CH                   ;プログラムの終了
         INT     21H
CODE     ENDS

;######### データ領域
DATA     SEGMENT
MSG      DB      '何かキーを押すとブザーがやみます.'
         DB      0DH,0AH,'$'
DATA     ENDS

;######### スタック領域
STACK    SEGMENT STACK
         DW      80H DUP (?)
STACK    ENDS
         END     START
```

```
;##########################################################
;#                                                        #
;#   ユーザー定義文字のフォントを登録する(PC-9800シリーズ互換)  #
;#                                                        #
;##########################################################
CODE     SEGMENT
         ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
         MOV     AX,SEG DATA             ;データセグメントを設定
         MOV     DS,AX
;
         MOV     BX,SEG PATARN           ;ユーザー定義文字のフォント登録
         MOV     CX,OFFSET PATARN
         MOV     DX,7621H
         MOV     AH,1AH
         INT     18H
;
         MOV     DX,OFFSET MSG           ;登録した文字を表示
         MOV     AH,09H
         INT     21H
EXIT:
         MOV     AH,4CH                  ;プログラムを終了
         INT     21H
CODE     ENDS

;#########   データ領域
DATA     SEGMENT
PATARN   DB      ?,?
         DB      16 DUP (10110011B,10001111B)
MSG      DB      0EBH,9FH                ;シフトJISコード EB9FH
         DB      'を登録しました．$'
DATA     ENDS

;######### スタック領域
STACK    SEGMENT STACK
         DW      80H DUP (?)
STACK    ENDS

         END     START
```

```
;##########################################################
;#                                                        #
;#        ＫＣＧアクセスモードを設定する                      #
;#                                                        #
;##########################################################
CODE     SEGMENT
         ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
         MOV     AX,SEG DATA             ;データセグメントを設定
         MOV     DS,AX
;
         MOV     AH,09H                  ;メッセージを表示
         MOV     DX,OFFSET MSG
         INT     21H
;
         MOV     AH,01H                  ;キー入力待ち
         INT     21H
;
         MOV     AH,1BH                  ;ドットアクセスを設定
         MOV     AL,01H
         INT     18H
;
         MOV     AH,01H                  ;キー入力待ち
         INT     21H
;
         MOV     AH,1BH                  ;コードアクセスを設定
         MOV     AL,0H
         INT     18H
EXIT:
         MOV     AH,4CH                  ;プログラムを終了
         INT     21H
CODE     ENDS
;#########   データ領域
DATA     SEGMENT
MSG      DB      '何かキーを押すと,ドットアクセスモードになり,'
         DB      0DH,0AH,0DH,0AH
         DB      '再度キーを押すと,コードアクセスモードに戻ります.'
         DB      0DH,0AH,0DH,0AH
         DB      'ドットアクセスモードでは漢字の表示はできません,'
         DB      0DH,0AH,'$'
DATA     ENDS

;######### スタック領域
STACK    SEGMENT STACK
         DW      80H DUP (?)
STACK    ENDS

         END     START
```

```
;#####################################################################
;#                                                                   #
;#      C R T  B I O S  の初期化とテキストカーソルの形状設定          #
;#                                                                   #
;#####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,1CH                  ;CRT BIOS の初期化
        MOV     AL,ØC1H                 ;    カーソルの点灯周期と
        MOV     DH,8                    ;    形状を変更
        MOV     DL,ØEH
        INT     18H
;
        MOV     AH,9                    ;メッセージの表示
        MOV     DX,OFFSET MSG
        INT     21H
;
        MOV     AH,1                    ;キー入力待ち
        INT     21H
;
        MOV     AH,1CH                  ;カーソルの点灯周期と
        MOV     AL,ØCCH                 ;    形状を標準にもどす
        MOV     DH,Ø
        MOV     DL,ØEH
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  データ領域
DATA    SEGMENT
MSG     DB      '何かキーを押すと,標準の状態に戻ります.'
        DB      ØDH,ØAH,'$'
DATA    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS
        END     START
```

```
;#####################################################################
;#                                                                   #
;#      V R A Mの画面幅(カラム数)を設定する                           #
;#                                                                   #
;#####################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,1DH                  ;画面幅を12Øカラムに設定
        MOV     AL,12Ø
        INT     18H
;
        MOV     AH,9                    ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
        MOV     AH,1                    ;キー入力待ち
        INT     21H
        MOV     AH,1DH                  ;画面幅を8Øカラムに設定
        MOV     AL,8Ø
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  データ領域
DATA    SEGMENT
MSG     DB      1AH
        DB      'Ø---*----1----*----2----*----3----*----4'
        DB      '---*----5----*----6----*----7----*----8'
        DB      '----*----9----*----A----*----B----*----'
        DB      ØDH,ØAH,ØDH,ØAH
        DB      '何かキーを押すと, 標準の状態に戻ります.'
        DB      ØDH,ØAH,'$'
DATA    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS
        END     START
```

```
;##################################################################
;#                                                               #
;#      テキストカーソルの形状を変更する                              #
;#                                                               #
;##################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,1EH                  ;カーソルタイプの設定
        MOV     AL,41H                  ;    ブリンクなしを指定
        MOV     DH,ØDH                  ;    カーソル形状を変更
        MOV     DL,ØEH
        INT     18H
;
        MOV     AH,9                    ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
        MOV     AH,1                    ;キー入力待ち
        INT     21H
;
        MOV     AH,1EH                  ;カーソルタイプを戻す
        MOV     AL,ØCCH                 ;    ブリンクありを指定
        MOV     DH,Ø                    ;    カーソル形状を戻す
        MOV     DL,ØEH
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT
MSG     DB      '何かキーを押すと,カーソルが標準の状態になります.'
        DB      ØDH,ØAH,'$'
DATA    ENDS

;######### スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                               #
;#      24ドットフォントパターンを読み出す                            #
;#                                                               #
;##################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,1FH                  ;"猫"のフォントを読む
        MOV     BX,OFFSET BUFF_G
        MOV     DX,472DH                ;    JISコードをセット
        INT     18H
;                                       ;フォントを表示する
        XOR     SI,SI                   ;{
;                                       ;  24行分繰り返し
        MOV     CX,24                   ;  {
        PUSH    CX
;                                       ;    1行分(24ドット)繰返し
RAW:    MOV     CX,3                    ;    {
        PUSH    CX
;                                       ;      8ドット分繰返し
DOT8:   MOV     BX,OFFSET BUFF_G        ;      {
        MOV     AL,[BX]+[SI]
        MOV     CX,8
CHECK:  ROL     AL,1                    ;        1ドット分表示
        JB      O
        JNB     X
X:      MOV     DX,OFFSET DOT_OFF       ;        Øなら'■'
        JMP     PUT
O:      MOV     DX,OFFSET DOT_ON        ;        1なら'　'
PUT:    PUSH    AX                      ;        1文字表示
        MOV     AH,9
        INT     21H
        POP     AX
        LOOP    CHECK
        POP     CX                      ;      }
        DEC     CX
        JZ      CRLF
        INC     SI
        PUSH    CX
        JMP     DOT8
CRLF:   MOV     AH,9                    ;      改行
        MOV     DX,OFFSET FEED
        INT     21H                     ;    }
        POP     CX
        DEC     CX
        JZ      EXIT
        INC     SI
        PUSH    CX
        JMP     RAW                     ;  }
```

```
EXIT:
        MOV     AH,4CH                   ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT
MSG     DB      '"猫"のフォントパターンを示します.',ØDH,ØAH,'$'
DOT_ON  DB      '■$'
DOT_OFF DB      ' $'
FEED    DB      ØDH,ØAH,'$'
BUFF_G  DB      72 DUP (?)
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;#############################################################
;#                                                           #
;#      メモリスイッチの内容を読む                           #
;#                                                           #
;#############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;                                       ;スイッチ(1〜8)分繰返し
        MOV     AL,1                    ;{
DSP_SW: MOV     AH,21H                  ;  ;メモリスイッチを読む
        INT     18H
;
        PUSH    AX
        ADD     AL,3ØH
        MOV     SW,AL
        MOV     SI,8                    ;  ;メモリスイッチを編集
LOOP:   SHR     DL,1
        JNB     BITØ
BIT1:   MOV     BIT[SI],'1'
        JMP     LOOP_E
BITØ:   MOV     BIT[SI],'Ø'
LOOP_E: DEC     SI
        JNZ     LOOP
        MOV     AH,9                    ;  ;メモリスイッチを表示
        MOV     DX,OFFSET MSG
        INT     21H                     ;}
;
        POP     AX
        INC     AL                      ;スイッチ番号 +1
        CMP     AL,8
        JNG     DSP_SW
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT
PROMPT  DB      'メモリスイッチの番号は (1-8): $'
MSG     DB      ØDH,ØAH,'メモリスイッチ [ '
SW      DB      ?
        DB      ' ] の内容は、'
BIT     DB      '"xxxxxxxx"'
        DB      ' です. ',ØDH,ØAH,'$'
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;###############################################################
;#                                                             #
;#      B e e p 音を設定する（周波数の設定）                       #
;#                                                             #
;###############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     HZ,400                  ;周波数の初期値=1000HZ
        MOV     CX,30
LOOP:   MOV     AH,23H                  ;スピーカの周波数を設定
        MOV     DX,HZ
        INT     18H
;
        MOV     AH,2                    ;BEEP音を鳴らす
        MOV     DL,7
        INT     21H
;
        MOV     AX,HZ                   ;周波数を変更
        ADD     AX,40
        MOV     HZ,AX
        DEC     CX
        JNE     LOOP
;
        MOV     AH,23H                  ;スピーカの周波数を設定
        MOV     DX,800                  ;周波数=800HZ
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  データ領域
DATA    SEGMENT
HZ      DW      ?
DATA    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS
        END     START
```

```
;###############################################################
;#                                                             #
;#      B e e p 音の周波数・鳴動時間を設定して，鳴らす              #
;#                                                             #
;###############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
START:
        MOV     AH,24H                  ;BEEP音の設定と起動
        MOV     DX,440                  ;周波数    =440HZ
        MOV     CX,100                  ;時間初期値 =100MSEC
        INT     18H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

# 第4章

# グラフBIOS

●グラフBIOS機能一覧

(INT 1DH)

| AHレジスタ | 機　　能 | |
|---|---|---|
| 00H | グラフBIOSの初期化 | (GINIT) |
| 01H | モード設定 | (GSCREEN) |
| 02H | 描画領域の指定 | (GVIEW) |
| 03H | 背景色等の指定 | (GCOLOR 1) |
| 04H | パレット番号と表示色の対応 | (GCOLOR 2) |
| 05H | 描画領域の塗りつぶし | (GCLS) |
| 06H | ドットの書き込み | (GPSET) |
| 07H | 直線，矩形の描画 | (GLINE) |
| 08H | 円，楕円の描画 | (GCIRCLE) |
| 09H | 指定色による塗りつぶし | (GPAINT 1) |
| 0AH | タイルパターンによる塗りつぶし | (GPAINT 2) |
| 0BH | 描画情報の格納 | (GGET) |
| 0CH | 描画情報を戻す | (GPUT1) |
| 0DH | 日本語の描画 | (GPUT 2) |
| 0EH | 描画画面の移動 | (GROLL) |
| 0FH | 指定ドットのパレット番号の取得 | (GPOINT 2) |
| 10H | 表示領域の設定 | (GSCROLL) |
| 11H | グラフィック画面の表示開始 | (GSTART) |
| 12H | グラフィック画面の表示停止 | (GSTOP) |
| 13H | グラフBIOSの終了 | (GTERM) |

注：PC-98XL/XAのグラフBIOSは，PC-9800シリーズのグラフBIOS，グラフLIOに相当する機能を持っているが，機能の呼び出し方法，およびG-VRAMの違いによるパラメータの内容が変更されている．使用に当っては十分注意すること．

## 4.1　グラフBIOS共通情報

・図形描画機能

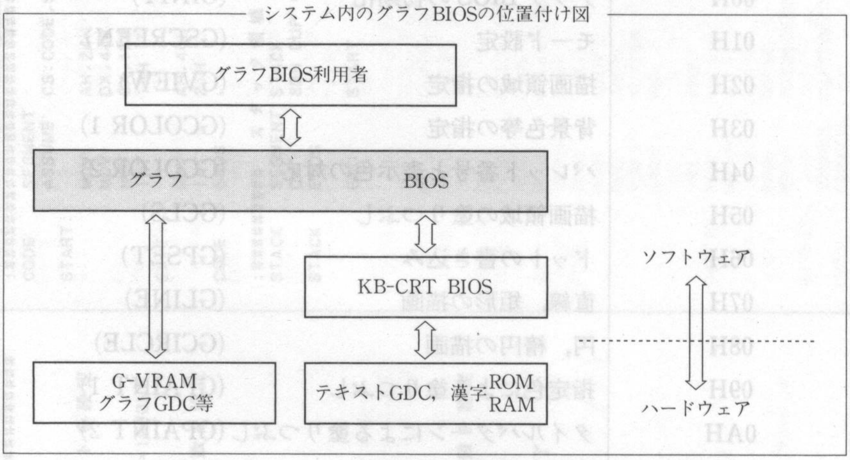
直線，四辺形，楕円，領域の塗りつぶし等の基本的な図形の描画を統一的な論理座標系をインターフェイスとして実現する．

・表示制御機能

カラー／白黒の表示モードに従い，512KB VRAM上のどの領域を表示するかを制御する．また，表示カラー，画面合成等の制御を行う．

・描画領域制御機能

画面モードに従って，表示制御とは独立に，図形等を描画する領域の制御を行う．




### (1) ユーザーインターフェイス

グラフBIOSでは，G-VRAM等のグラフィック用ハードウェアを論理的な形でユーザーに提供する．

① 画面モード

画面モード：カラー，白黒

分　解　能：1120×936ドット(ただし，ディスプレイに表示できるのは1120×750ドット)

② アクティブ画面

a）カラーモード

512Kバイトすべて使用…………1ページ(ページ0)

b）白黒モード

512Kバイトを4分割して使用…4ページ(ページ0～3)

③ ディスプレイ画面

a）カラーモード

0……表示しない.

1……表示する.

b）白黒モード

0……表示しない.

1……ページ 0 を表示.

2……ページ 1 を表示.

3……ページ 0 とページ 1 の合成表示.

4……ページ 2 を表示.

5……ページ 0 とページ 2 の合成表示.

6……ページ 1 とページ 2 の合成表示.

7……ページ 0, 1, 2 の合成表示.

8……ページ 3 を表示.

9……ページ 0 とページ 3 の合成表示.

10……ページ 1 とページ 3 の合成表示.

11……ページ 0, 1, 3 の合成表示.

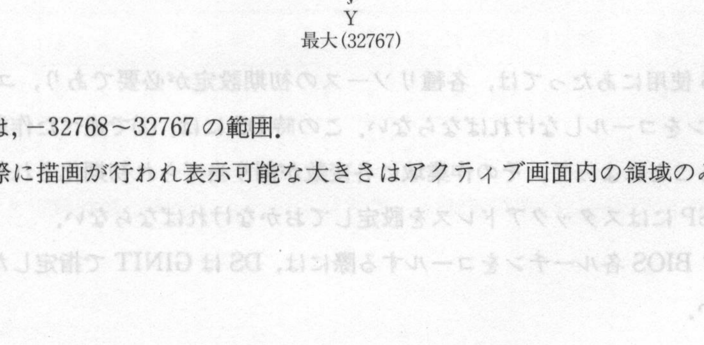
12……ページ 2 とページ 3 の合成表示.

13……ページ 0, 2, 3 の合成表示.

14……ページ 1, 2, 3 の合成表示.

15……ページ 0, 1, 2, 3 の合成表示.

④ 論理座標系




指定座標値は, −32768～32767 の範囲.

ただし, 実際に描画が行われ表示可能な大きさはアクティブ画面内の領域のみ.

⑤ カラー指定

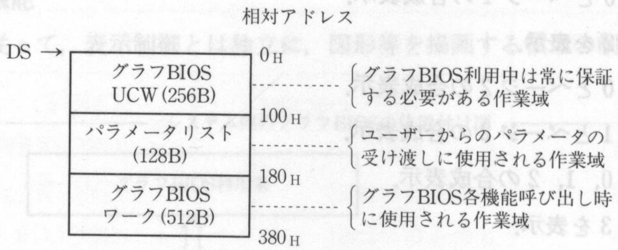
図形描画時のカラー指定はパレット番号により行う．

対応
表示色 0～4095 ⟷ パレット番号 0～15

白黒モード時は，パレット番号 0…………黒パレットレジスタで指定される表示色

〃　　　　0 以外……白　　　　〃

### (2) 作業域




使用スタックの大きさ　グラフBIOS部で64バイト(＋20)

### (3) 注意事項

グラフ BIOS は，G-VRAM，グラフ GDC を一括して制御しているため，グラフ BIOS 使用時にユーザーが直接 G-VRAM，グラフ GDC をアクセスする場合は，グラフ BIOS の制御と矛盾を生じないよう細心の注意が必要である．

一般にグラフ BIOS 使用時は，ユーザーはグラフ BIOS を通してのみ G－VRAM，グラフ GDC にアクセスする方が良い．

ユーザーが直接 G-VRAM，グラフ GDC へアクセスする場合には，グラフ BIOS の管理情報と矛盾を生じない様に注意するだけでなく，ハード上の制約事項も十分に熟知している必要がある．

## 4.1.1　使用方法

グラフ BIOS の機能を利用するにあたっての使用方法および制約事項等を次に述べる．

### (1) 初期設定

グラフ BIOS 使用にあたっては，各種リソースの初期設定が必要であり，ユーザーは必ず GINIT ルーチンをコールしなければならない．この時 DS には，(2)で述べた作業域を示すアドレスを設定することによって，その作業域に各変数が割り当てられ初期化される．

また，SS，SP にはスタックアドレスを設定しておかなければならない．

以降，グラフ BIOS 各ルーチンをコールする際には，DS は GINIT で指定した値と同一でなければならない．

### (2) 呼び出し方法

グラフBIOS各ルーチンを呼び出す時には，次の様に行うこと．

① DSにGINIT時指定したDSと同じ値を設定する．

② 必要ならばパラメータリストを作成する．また，必要ならば各レジスタにパラメータを設定する．

③ グラフBIOSで使用可能なスタックアドレスをSS，SPに設定する．

④ 対応するルーチンの機能番号をAHに格納して，INT 1DHを実行する．

### (3) 注意事項

①グラフBIOS処理中は，外部割り込み可となっている．

② BIOSルーチンは，必ずシリアルに呼び出さなければならない．

③ 指定値以外の値をパラメータとして指定した場合，結果は保証されない．

④ グラフBIOSでは，比較的時間がかかる描画処理を行っている時(GCIRCLE，GPAINT1/2など)，描画処理の中断を可能とするために，一定処理ごとに，GINITで渡されたアドレスへFar Callを行っている．

このため，ユーザーは，このアドレスに何らかの対応するルーチン(Far RETのみのルーチンでも良い)のアドレスを設定しておかなければならない．この時，このアドレスに置かれるルーチンは，次のインターフェイスを守らなくてはならない．

・セグメントレジスタも含めて全レジスタを保証しなくてはならない．

・このルーチンから再びグラフBIOSルーチンを呼び出してはならない．

・グラフBIOSルーチン使用の作業域をこわしてはならない．

・グラフBIOS管理リソース(G-VRAM，グラフGDC等)の状態を変更してはならない．

ただし，グラフBIOSの処理続行が不要な場合には，グラフBIOSルーチンに制御を戻す必要はない．

⑤ リターン時のパラメータリストの内容は保証される．

## 4.2　グラフBIOSの初期化(GINIT)

### (1) 機能

グラフBIOSの初期化を行う．

グラフBIOSを使用する場合には，必ず最初にこのコマンドを実行する必要がある．このコマンドを実行することによって，

・画面モードはカラーグラフィックモード．

・アクティブ画面(描画対象となる領域，ページとも呼ぶ)は0．

・ディスプレイ画面(表示モードに従って表示される領域)は1．

・パレット番号は次表のとおり．

| パレット番号 | 色 | G | | | | R | | | | B | | | | 16進表示 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $2^3$ | $2^2$ | $2^1$ | $2^0$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | | | |
| 0 | 黒 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 暗い青 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | A |
| 2 | 暗い赤 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | A | 0 |
| 3 | 暗い紫 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | A | A |
| 4 | 暗い緑 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | A | 0 | 0 |
| 5 | 暗い水色 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | A | 0 | A |
| 6 | 暗い黄色 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | A | A | 0 |
| 7 | 暗い白 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | A | A | A |
| 8 | 灰 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 7 | 7 | 7 |
| 9 | 青 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | F |
| 10 | 赤 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | F | 0 |
| 11 | 紫 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | F | F |
| 12 | 緑 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | F | 0 | 0 |
| 13 | 水色 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | F | 0 | F |
| 14 | 黄色 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | F | F | 0 |
| 15 | 白 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | F | F | F |

アクティブページ内描画領域は，アクティブページ全体に初期設定される．

**(2) 入力**

・内部割り込みコード← 1DH

・AH ← 00H

・DS ←グラフ BIOS ワークエリア(ユーザー UCW)のセグメントベース

グラフ BIOS が使用するワークエリア 1380H バイトはユーザーが確保する.

・DS : 100H～DS : 101H ←中断処理ルーチンオフセットアドレス

・DS : 102H～DS : 103H ←中断処理ルーチンセグメントアドレス

・SS/SP ←ユーザーが確保するグラフ BIOS スタックエリア

グラフ BIOS スタックエリア約 64 バイトはユーザーが確保する.

**(3) 出力**

AX 以外のすべてのレジスタが保証される.

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

**(4) 注意事項**

描画領域の初期値は(0, 0) − (1119, 935)となる.

表示領域の表示開始 Y 座標は 0 となる.

画面スイッチが 02H に設定されるため, グラフィックを描画しても表示されない. 表示させるためには, GSCREEN で画面スイッチを 00H または 01H に変更するか, GSTART を呼ぶ必要がある.

## 4.3　モード設定(GSCREEN)

### (1) 機能

画面モード，画面スイッチ，アクティブ画面，ディスプレイ画面を設定する．

本コマンドによって，描画領域はアクティブ画面全体となり，表示領域の表示開始Y座標は初期化される．

モードが変更された時のパレットレジスタは，前のモードでの値が使用される．

### (2) 入力

・内部割り込みコード← 1DH

・AH ← 01H

・DS ←グラフ BIOS ワークエリア(ユーザー UCW)のセグメントベース(GINIT と同じ値)

・パラメータリスト(詳細は(4)を参照)

| DS+100H | +101H | +102H | +103H |
|---|---|---|---|
| 画面モード<br>(表示モード) | 画面スイッチ | アクティブ画面 | ディスプレイ<br>画面 |

### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

注：画面モード，アクティブ画面，ディスプレイ画面の組合せが不適当な場合は，処理を行わずエラーリターンする．

### (4) パラメータリストの詳細

① 画面モード

画面モードは次の2種類がある．

| パラメータ | 画面モード名称 |
|---|---|
| 00H | カラーグラフィックモード |
| 01H | モノクログラフィックモード |
| FFH | 省略，今までの画面モードを引継ぐ |

② 画面スイッチ

| パラメータ | グラフィック画面の表示有無 | 高速描画有無 |
|---|---|---|
| 00H | グラフィック画面表示　有 | 普通描画 |
| 01H | グラフィック画面表示　有 | 高速描画 |
| 02H | グラフィック画面表示　無 | 普通描画 |
| 03H | グラフィック画面表示　無 | 高速描画 |
| FFH | 省略，今までの画面スイッチの状態を引継ぐ | |

③ アクティブ画面

| パラメータ | アクティブ画面 |
|---|---|
| 00H | ページ0（カラーはページ0のみ） |
| 01H | ページ1 |
| 02H | ページ2 |
| 03H | ページ3 |
| FFH | 省略．今までのアクティブ画面を引継ぐ．ただし，画面モード変更時はページ0をアクティブ画面とする． |

④ ディスプレイ画面

b）グラフィックモード

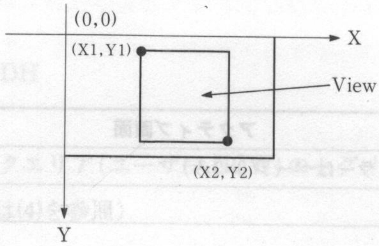
| パラメータ | 表示画面 |
|---|---|
| 0 | 表示しない |
| 1 | 画面0を表示（カラーは画面Dのみ） |
| 2 | 画面1を表示 |
| 3 | 画面0と1を合成して表示 |
| 4 | 画面2を表示 |
| 5 | 画面0と2を合成して表示 |
| 6 | 画面1と2を合成して表示 |
| 7 | 画面0，1，2を合成して表示 |
| 8 | 画面3を表示 |
| 9 | 画面0と3を合成して表示 |
| 10 | 画面1と3を合成して表示 |
| 11 | 画面0，1，3を合成して表示 |
| 12 | 画面2と3を合成して表示 |
| 13 | 画面0，2，3を合成して表示 |
| 14 | 画面1，2，3を合成して表示 |
| 15 | 画面0，1，2，3を合成して表示 |
| 255(FFH) | 省略．今までのディスプレイページを引き継ぐ．ただし，画面モード変更時は01Hとみなす． |

## 4.4　描画領域の指定(GVIEW)

### (1) 機能

アクティブ画面内の描画領域(View Port：ビューポート)を指定する．またビューポート内の塗りつぶし，外枠の描画を行う．

このコマンドの実行によって，アクティブ画面への図形描画は，ビューポート内にのみ反映されるようになる．



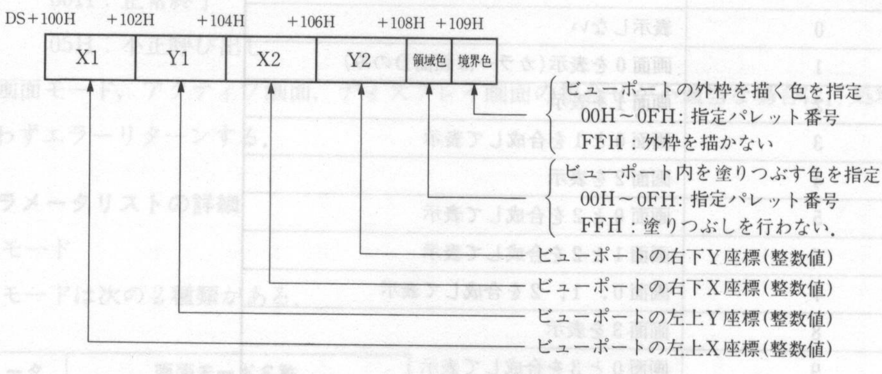

### (2) 入力

・内部割り込みコード←1DH

・AH←02H

・DS←グラフBIOSワークエリア(ユーザーUCW)のセグメントベース(GINITと同じ値)

・パラメータリスト




注：X1<X2，Y1<Y2の関係をもつこと．
(X1，Y1)，(X2，Y2)はアクティブ画面上の座標であること．

### (3) 出力

AX以外のすべてのレジスタが保証される．

・AH←終了条件

00H：正常終了

05H：不正呼び出し

## 4.5　背景色等の指定(GCOLOR 1)

### (1) 機能

バックグラウンドカラー，フォアグラウンドカラーを指定する．

バックグラウンドカラーとは，グラフィック画面の地のパレット番号のことで，この命令実行後 GCLS 命令によって画面をクリアすると，この色によって画面が塗り変えられる．また，以後 GPSET 命令を色指定なしで実行すると，この色が採用される．

フォアグラウンドカラーとは，図形描画においてパレット番号省略時に使用されるパレット番号のことである．

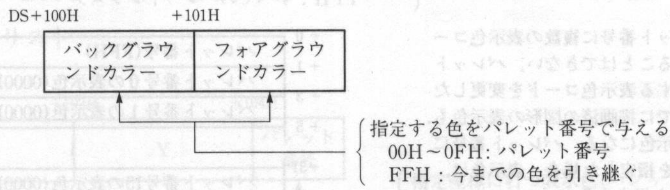
### (2) 入力

・内部割り込みコード← 1DH

・AH ← 03H

・DS ←グラフ BIOS ワークエリアのセグメントベース

・パラメータリスト




### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

## 4.6　パレット番号と表示色の対応(GCOLOR2)

### (1) 機能

パレット番号と表示色コードを対応させ，パレット番号で表わす色を定義する．

### (2) 入力

・内部割り込みコード← 1DH

・AH ← 04H

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト

| DS+100H | +101H |
|---|---|
| パレット番号 | 表示色コード |

表示色コード：表示色　0000H～0FFFH

パレット番号：表示色を設定するパレット番号
00H～07H：表示色を設定するパレット番号
80H：黒パレットレジスタ
81H：白パレットレジスタ
FFH：すべてのパレットレジスタ

注：1つのパレット番号に複数の表示色コードを指定することはできない．パレット番号の対応する表示色コードを変更した場合は，すでに描画済の図形の表示色も，変更した表示色になる．パレット番号に80H，81Hを指定した場合，表示色は，0～FFFFHを指定する．

| オフセット | 内容 |
|---|---|
| +0 | パレット番号(FFH) |
| +1 | パレット番号 0 の表示色(0000H～0FFFH) |
| +3 | パレット番号 1 の表示色(0000H～0FFFH) |
| +5 | ～ |
| +31 | パレット番号15の表示色(0000H～0FFFH) |
| +33 | |

### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

## 4.7　描画領域の塗りつぶし(GCLS)

### (1) 機能

アクティブ画面内の描画領域を，バックグラウンドカラーのパレット番号の表示色で塗りつぶす．

### (2) 入力

・内部割り込みコード← 1DH

・AH ← 05H

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

**(3) 出力**

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

## 4.8 ドットの書き込み(GPSET)

**(1) 機能**

アクティブページ内の描画領域内の指定の座標に，指定の色で点を打つ(描く)．色はパレット番号で指定する．

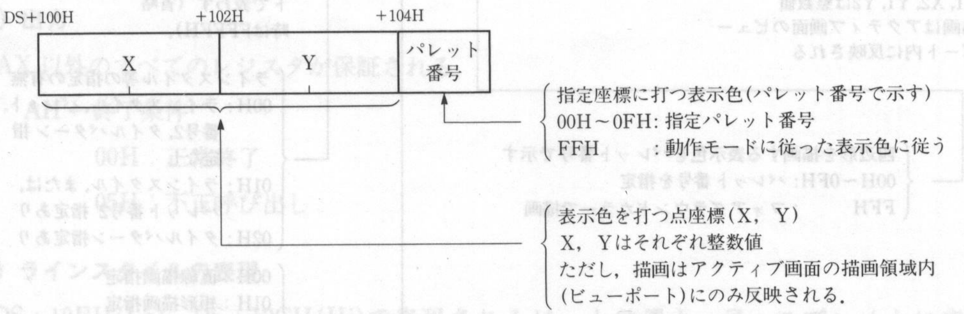
**(2) 入力**

・内部割り込みコード← 1DH

・AH ← 06H

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




・AL ←動作モードの指定

01H：表示色(パレット番号)省略時，ファアグラウンドカラーのパレット番号を使用する．

02H：表示色(パレット番号)省略時，バックグラウンドカラーのパレット番号を使用する．

**(3) 出力**

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

## 4.9　直線，矩形の描画(GLINE)

### (1) 機能

アクティブページのビューポート内の指定された2点を結ぶ線分，または，この線分を対角線とする矩形を描画する．矩形内を塗りつぶすこともできる．

### (2) 入力

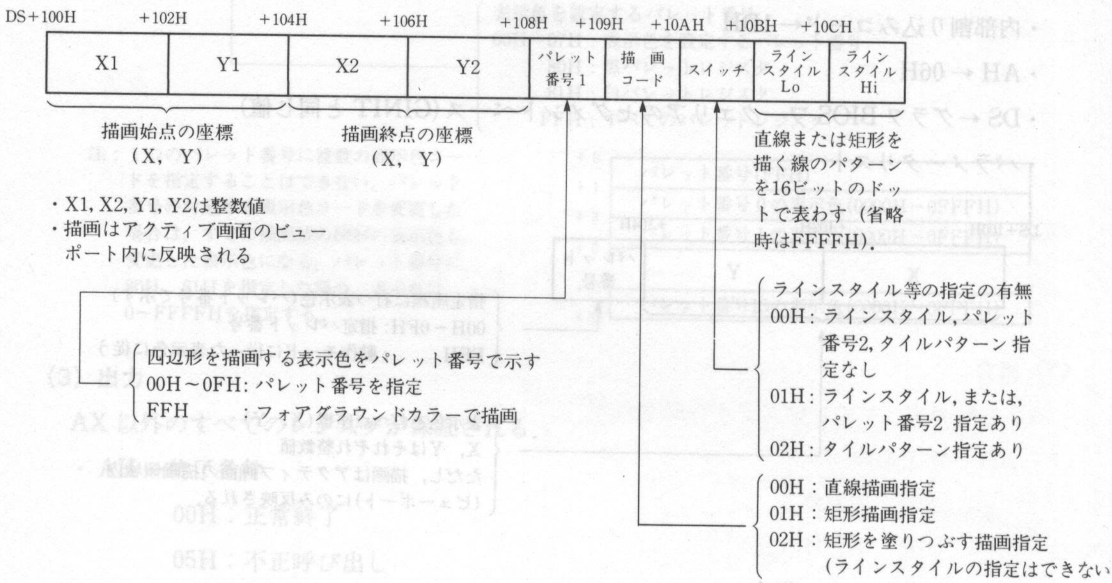
・内部割り込みコード←1DH

・AH←07H

・DS←グラフBIOSワークエリアのセグメントベース(GINITと同じ値)

・パラメータリスト

① 直線，矩形の場合



② 塗りつぶされた矩形の場合

| DS+100H | +102H | +104H | +106H | +108H | +109H | +10AH | +10BH | +10CH | +10DH | +10EH | +110H |
|---|---|---|---|---|---|---|---|---|---|---|---|
| X1 | Y1 | X2 | Y2 | パレット番号1 | 描画コード | スイッチ | パレット番号2 | ラインスタイルHi | タイルパターン長 | タイルパターンオフセット | タイルパターンベース |

オフセットアドレス　セグメントベースアドレス
(0000H~FFFFH)

(タイルパターン格納域)
タイルパターン格納域の形式は「4.12 GPAINT2」を参照すること.
(描画コードが02Hのときのみ有効)

タイルパターン長
00H~FFH：
タイルパターン格納域長
(描画コードが02Hのときのみ有効)

スイッチ：
ラインスタイル，四辺形塗りつぶし色，タイルパターンの指定スイッチ
00H：ラインスタイル，四辺形塗りつぶし色，タイルパターンの指定なし
01H：ラインスタイルまたは四辺形塗りつぶし色の指定あり
02H：タイルパターン指定あり
(描画コードが02Hで，四辺形塗りつぶし色もタイルパターンも指定しない場合，四辺形の内部は四辺形の描画色で塗りつぶされる.)

パレット番号2：
四辺形内部の塗りつぶし色
00H~0FH: 指定のパレット番号で四辺形の内部を塗りつぶす
(描画コードが02Hのときのみ有効)
描画コードが02H以外の場合はラインスタイルLoバイトとなる

注：特に説明のない部分は①と同じ

### (3) 出力

AX以外のすべてのレジスタが保証される.

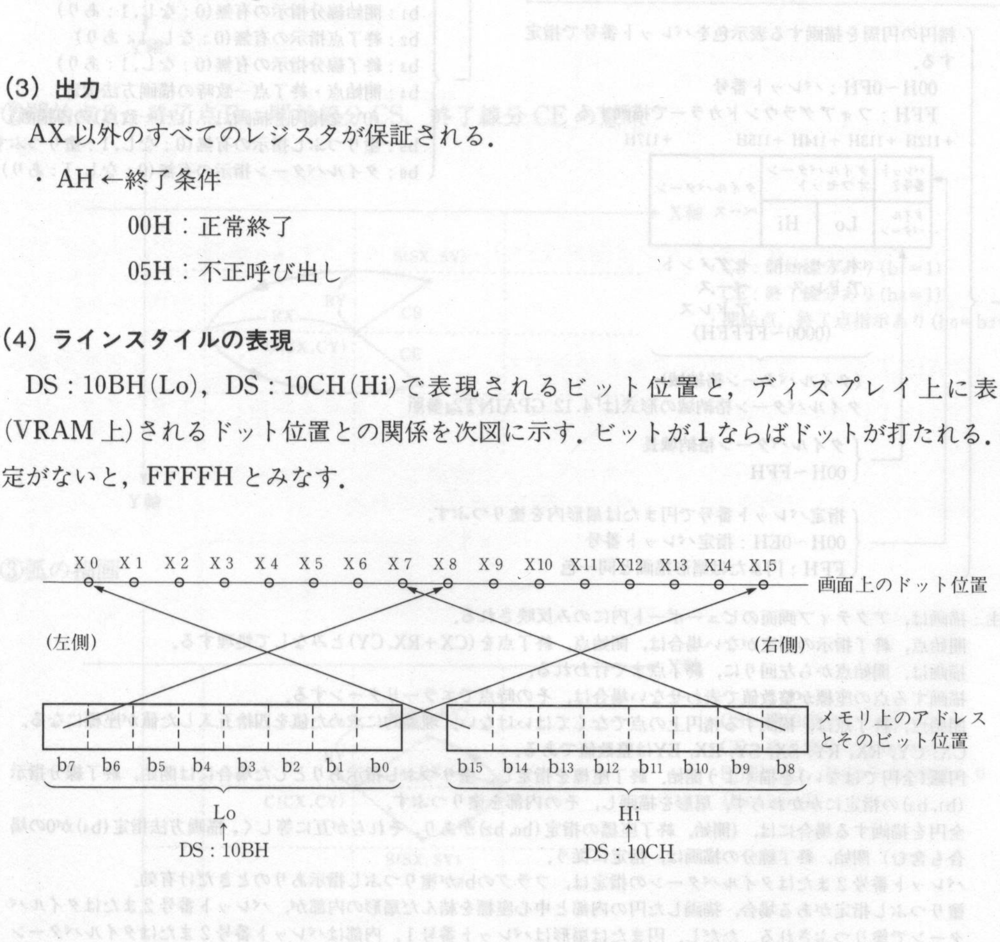
・AH←終了条件

00H：正常終了

05H：不正呼び出し

### (4) ラインスタイルの表現

DS：10BH(Lo)，DS：10CH(Hi)で表現されるビット位置と，ディスプレイ上に表現(VRAM上)されるドット位置との関係を次図に示す.ビットが1ならばドットが打たれる.指定がないと，FFFFHとみなす.



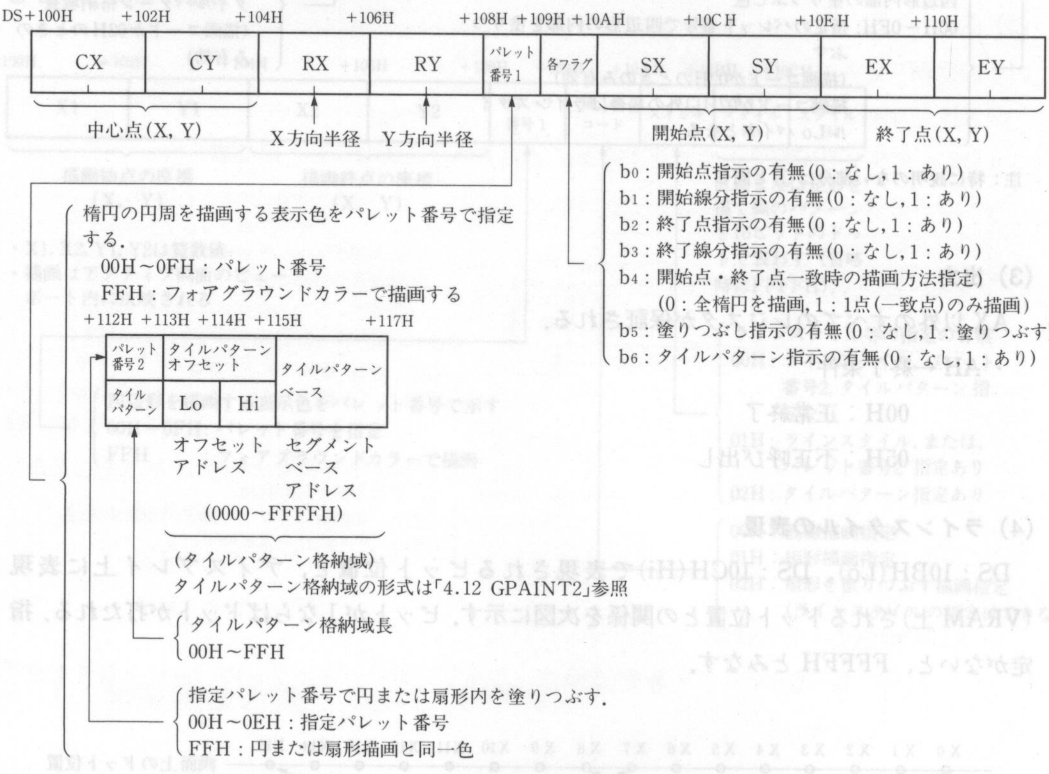
## 4.10 円，楕円の描画(GCIRCLE)

### (1) 機能

指定された中心点，X方向半径，Y方向半径をもとに，楕円または円を描画する．
開始点，終了点を指定することにより，円弧，または扇形を描画することができる．
円，楕円，扇形の内部を塗りつぶすことができる．

### (2) 入力

・内部割り込みコード← 1DH
・AH ← 08H
・DS ←グラフBIOSワークエリアのセグメントベース(GINITと同じ値)
・パラメータリスト




注：描画は，アクティブ画面のビューポート内にのみ反映される．
　開始点，終了指示の指定がない場合は，開始点，終了点を(CX+RX, CY)とみなして処理する．
　描画は，開始点から左回りに，終了点まで行われる．
　描画する点の座標が整数値で表わせない場合は，その時点でエラーリターンする．
　開始点，終了点は，描画する楕円上の点でなくてはいけない．理論的に求めた値を四捨五入した値が座標になる．
　CX, CY, RX, RY, SX, SY, EX, EYは整数値である．
　円弧(全円ではない)を描くよう開始．終了座標を指定し，塗りつぶし指示ありとした場合には開始，終了線分指示($b_1$, $b_3$)の指定にかかわらず，扇形を描画し，その内部を塗りつぶす．
　全円を描画する場合には，(開始，終了座標の指定($b_0$, $b_2$)があり，それらが互に等しく，描画方法指定($b_4$)が0の場合も含む) 開始，終了線分の描画は，指定に従う．
　パレット番号2またはタイルパターンの指定は，フラグの$b_5$が塗りつぶし指示ありのときだけ有効．
　塗りつぶし指定がある場合，描画した円の内部と中心座標を結んだ扇形の内部が，パレット番号2またはタイルパターンで塗りつぶされる．ただし，円または扇形はパレット番号1，内部はパレット番号2またはタイルパターン(省略時は，パレット番号1)で描画される．

**(3) 出力**

AX以外のすべてのレジスタが保証される．

・AH←終了条件

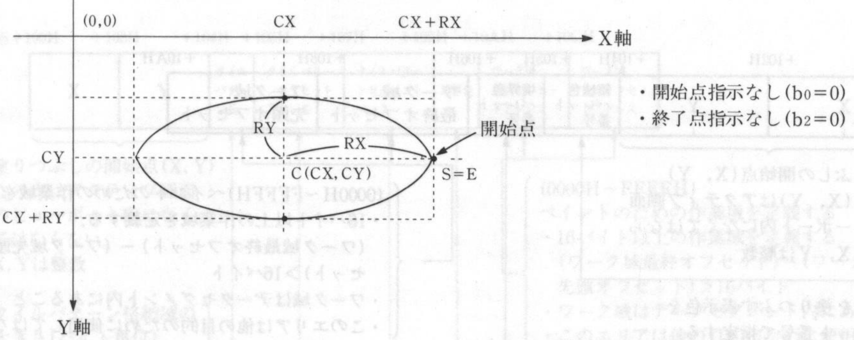
00H：正常終了

05H：不正呼び出し

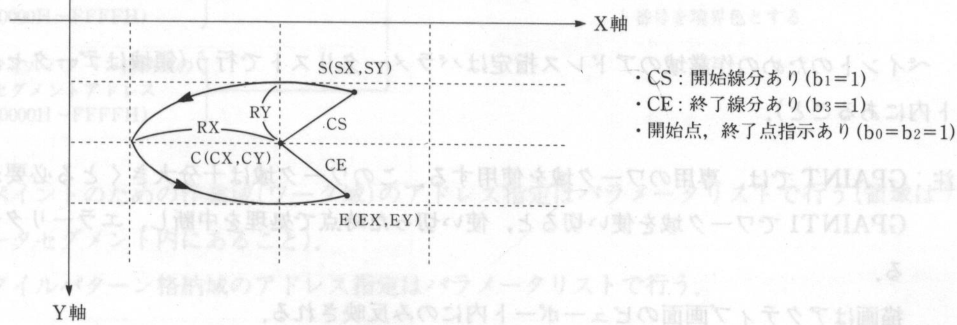
06H：演算オーバーフロー

**(4) パラメータの説明**

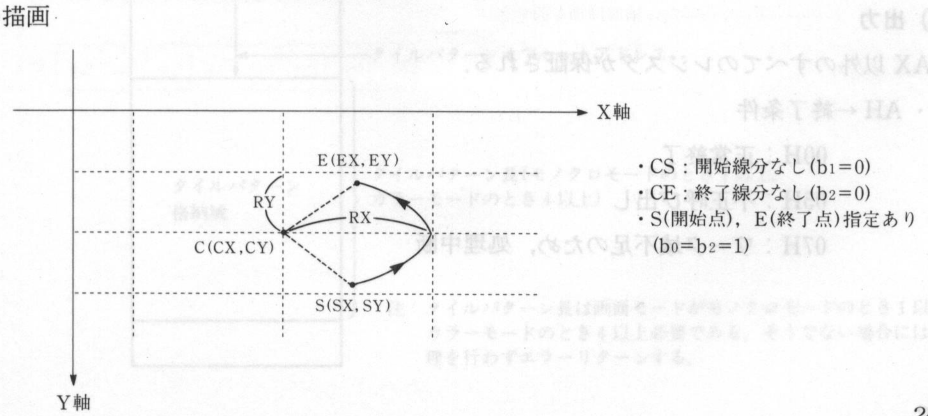
① 楕円（描画開始点，終了点を指定しない場合）




②開始点S，終了点E，開始線分CS，終了線分CEの意味




③弧の描画



## 4.11　指定色による塗りつぶし(GPAINT1)

### (1) 機能

指定した点と境界色で決定される領域を，指定の色(領域色になる)で塗りつぶす．

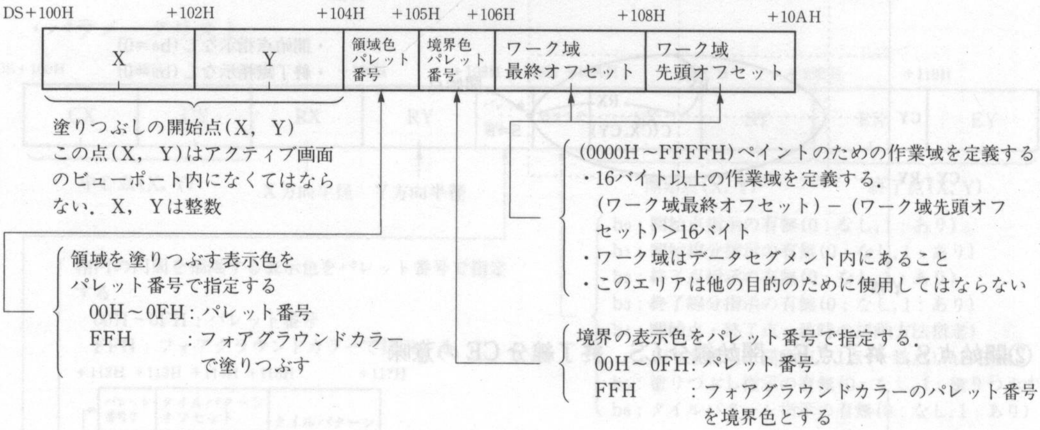
### (2) 入力条件

・内部割り込みコード←1DH

・AH←09H

・DS←グラフBIOSワークエリアのセグメントベース(GINITと同じ値)

・パラメータリスト




ペイントのための作業域のアドレス指定はパラメータリストで行う(領域はデータセグメント内にあること)．

注：GPAINTでは，専用のワーク域を使用する．このワーク域は十分大きくとる必要がある．GPAINT1でワーク域を使い切ると，使い切った時点で処理を中断し，エラーリターンする．

描画はアクティブ画面のビューポート内にのみ反映される．

### (3) 出力

AX以外のすべてのレジスタが保証される．

・AH←終了条件

00H：正常終了

05H：不正呼び出し

07H：ワーク域不足のため，処理中断

# 4.12 タイルパターンによる塗りつぶし(GPAINT2)

## (1) 機能

指定した点と境界色で決定される領域を，指定をタイルパターンで塗りつぶす．

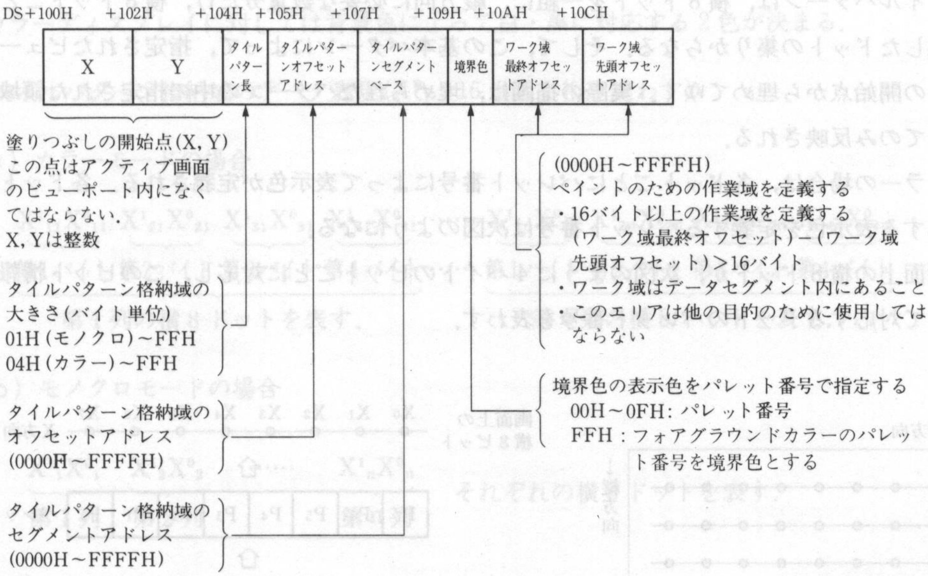
## (2) 入力

・内部割り込みコード← 1DH

・AH ← 0AH

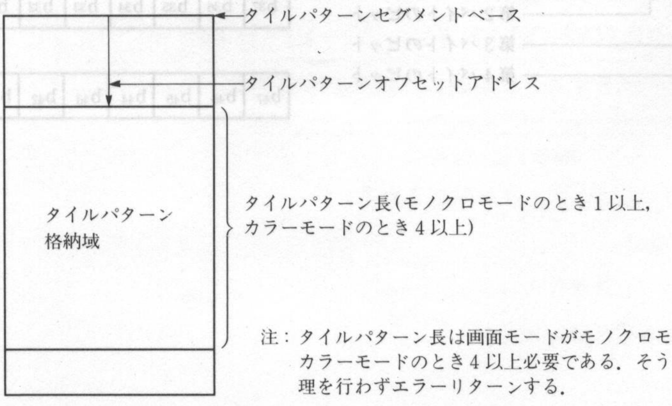
・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




・ペイントのための作業域(ワーク域)のアドレス指定はパラメータリストで行う(領域はデータセグメント内にあること)．

・タイルパターン格納域のアドレス指定はパラメータリストで行う．




注：タイルパターン長は画面モードがモノクロモードのとき1以上，カラーモードのとき4以上必要である．そうでない場合には，処理を行わずエラーリターンする．

**(3) 出力**

AX以外のすべてのレジスタが保証される.

・AH←終了条件

00H：正常終了

05H：不正呼び出し

07H：ワーク域不足のため，処理中断

**(4) タイルパターンの説明**

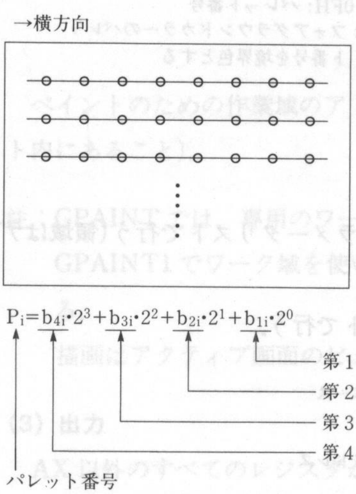
タイルパターンの格納域に格納するタイルパターン形式は次のようになる.

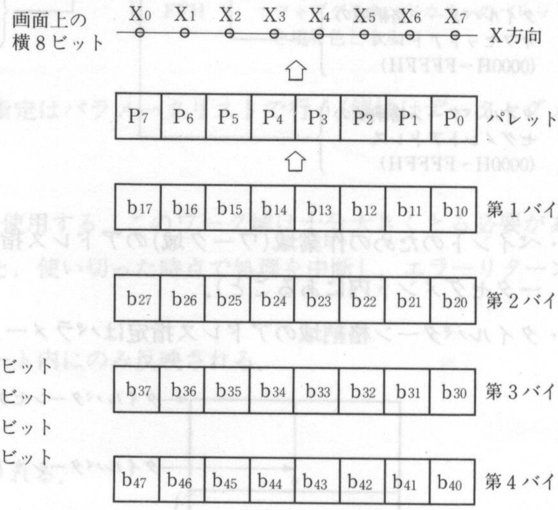
① 画面モードがカラーの場合

タイルパターンは，横8ドットを一組に，縦方向に必要な数量分だけ，横8ドットごとに定義したドットの集りからなる．そして，この基本パターンによって，指定されたビューポートの開始点から埋めてゆく．実際の描画は，埋められたパターンの中の指定された領域についてのみ反映される.

カラーの場合は，各ドットごとにパレット番号によって表示色が定義される．各ドットに対応する表示色を定義するパレット番号は次図のようになる.

画面上の横8ドットが，次図のように4バイトのビットごとに対応し，このビット情報によって対応するドットのパレット番号を表わす.

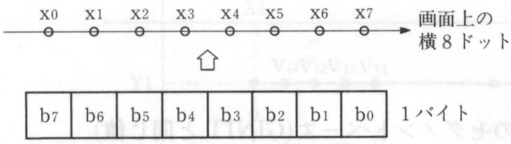


② 画面モードがモノクロの場合

タイルパターンは，横8ドットの白・黒表示を一組に，縦方向に必要な数量分だけ，横8ドットごとに定義したドットの集りからなる．

横8ドットの白・黒表示は1バイトの8ビット情報で表わす．




$b_i$が1ならば対応するドットは白で，$b_i$が0ならば対応するドットは黒で表示される．

カラーディスプレイに対しては背景色によって白・黒に対応する2色が決まる．

③ タイルパターン格納域のデータ表現($X^R{}_1$：16進数1桁を表わす)

a）カラーモードの場合

$X^1{}_{11}X^0{}_{11}$ $X^1{}_{21}X^0{}_{21}$ $X^1{}_{31}X^0{}_{31}$ $X^1{}_{41}X^0{}_{41}$ …… $X^1{}_{1n}X^0{}_{2n}$ $X^1{}_{2n}X^0{}_{2n}$ $X^1{}_{3n}X^0{}_{3n}$ $X^1{}_{4n}X^0{}_{4n}$

第1バイト 第2バイト 第3バイト 第4バイト …… 第1バイト 第2バイト 第3バイト 第4バイト

第1列の横8ドットを表す． 第n列の横8ドットを表す．

b）モノクロモードの場合

$X^1{}_1X^0{}_1$ $X^1{}_2X^0{}_2$ …… $X^1{}_nX^0{}_n$ それぞれの横8ドットを表す．

第1列 第2列 第n列

④ タイルパターンの描画位置

タイルをはりつめるのは，ビューポートの基点(左上)から始まる．

## 4.13　描画情報の格納(GGET)

### (1) 機能

指定領域の描画情報を，指定の格納域へ格納する.

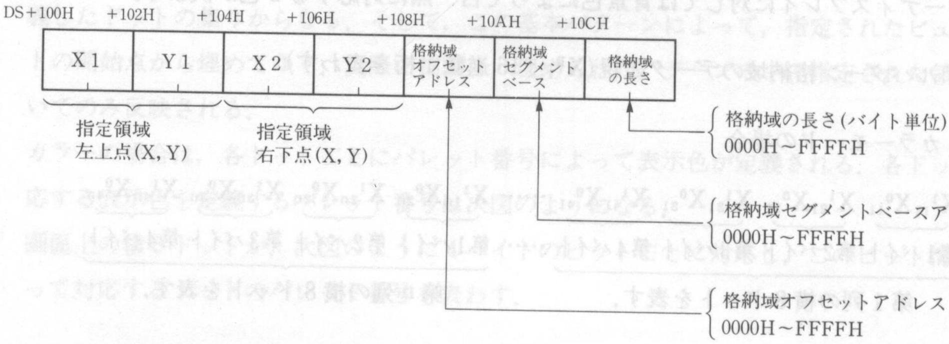
### (2) 入力条件

・内部割り込みコード← 1DH

・AH ← 0BH

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




① 指定する座標の条件

(X1, Y1)，(X2, Y2)はアクティブ画面上のビューポート内にあること.

X2≧X1, Y2≧Y1 であること.

X, Y は整数値.

② 格納域の指定条件(￥：整数の割り算の商(余り切り捨て), ・：乗算)

画面モードがカラーの場合

格納域の長さ≧((X2−X1+8)￥8)・(Y2−Y1+1)・4+4

画面モードがモノクロの場合

格納域の長さ≧((X2−X1+8)￥8)・(Y2−Y1+1)+4

格納域にパラメータリストで定義した領域を確保する.

### (3) 出力

AX 以外のすべてのレジスタが保証される.

・AH ←終了条件

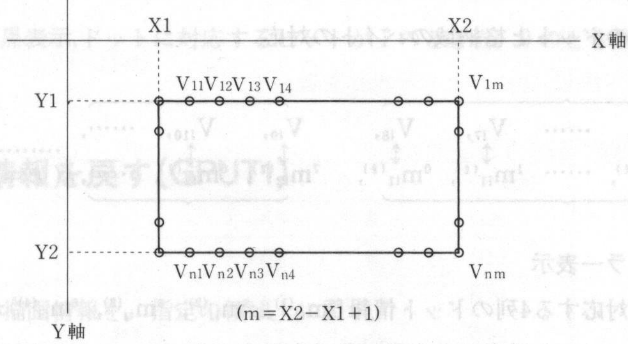
00H：正常終了

05H：不正呼び出し

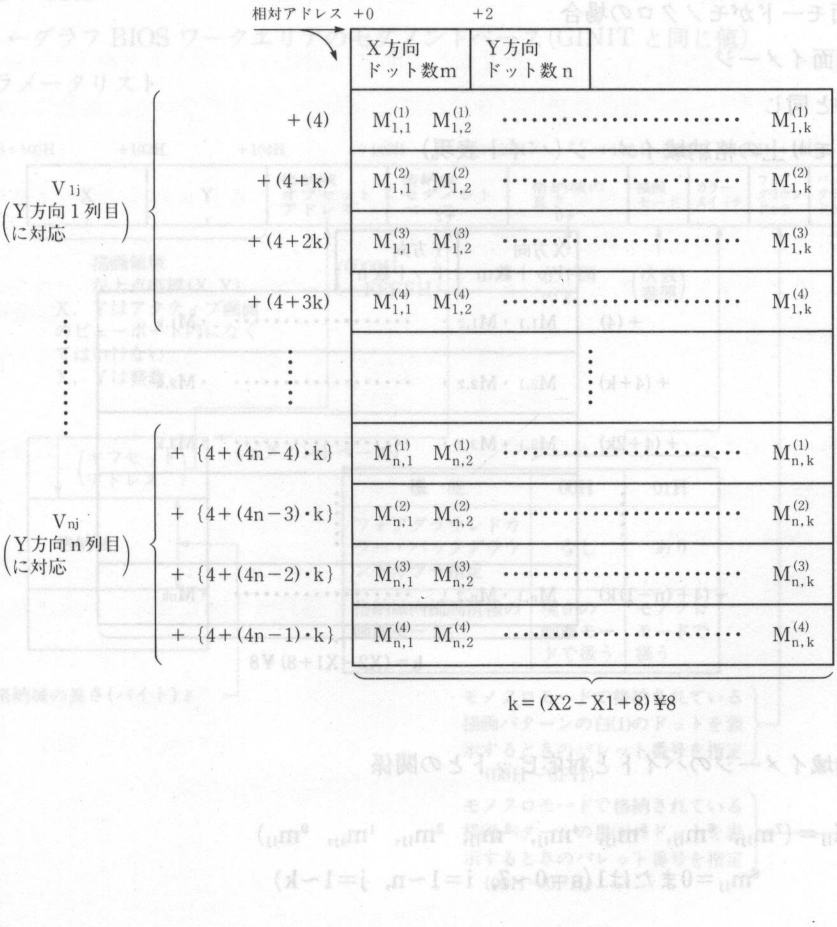
(4) 格納域の形式

①画面モードがカラーの場合

・画面イメージ




・メモリ上の格納域イメージ(バイト表現)



・格納域イメージのバイトと対応ビットとの関係

$$\begin{array}{cccccccc} 2^7\text{ビット}, & 2^6\text{ビット}, & 2^5\text{ビット} & \cdots\cdots & & & 2^1\text{ビット}, & 2^0\text{ビット} \\ \Downarrow & \Downarrow & \Downarrow & & & & \Downarrow & \Downarrow \end{array}$$

$$M_{ij}{}^{(\ell)} = ({}^{7}m_{ij}{}^{(\ell)},\ {}^{6}m_{ij}{}^{(\ell)},\ {}^{5}m_{ij}{}^{(\ell)},\ {}^{4}m_{ij}{}^{(\ell)},\ {}^{3}m_{ij}{}^{(\ell)},\ {}^{2}m_{ij}{}^{(\ell)},\ {}^{1}m_{ij}{}^{(\ell)},\ {}^{0}m_{ij}{}^{(\ell)})$$

${}^{s}m_{ij}{}^{(\ell)} = 0$ または $1$（$s = 0 \sim 7,\ i = 1 \sim n,\ j = 1 \sim k,\ l = 1 \sim 4$）

・画面イメージのドットと格納域のバイトの対応

$$\begin{array}{ccccccccc} V_{i1}, & V_{i2}, & \cdots\cdots & V_{i7}, & V_{i8}, & V_{i9}, & V_{i10}, & \cdots\cdots, & \\ \updownarrow & \updownarrow & & \updownarrow & \updownarrow & \updownarrow & \updownarrow & & \cdots\cdots \\ {}^{7}m_{i1}{}^{(\ell)}, & {}^{6}m_{i1}{}^{(\ell)}, & \cdots\cdots & {}^{1}m_{i1}{}^{(\ell)}, & {}^{0}m_{i1}{}^{(\ell)}, & {}^{7}m_{i2}{}^{(\ell)}, & {}^{6}m_{i2}{}^{(\ell)}, & \cdots\cdots, & (\ell=1,2,3,4) \end{array}$$

・各ドットのカラー表示

あるドットに対応する4列のドット情報（${}^{s}m_{ij}{}^{(1)},\ {}^{s}m_{ij}{}^{(2)},\ {}^{s}m_{ij}{}^{(3)},\ {}^{s}m_{ij}{}^{(4)}$）から，次の式によって計算されるパレット番号が示す表示色で表示される．

$$P = {}^{s}m_{ij}{}^{(1)} + 2 \cdot {}^{s}m_{ij}{}^{(2)} + 4 \cdot {}^{s}m_{ij}{}^{(3)} + 8 \cdot {}^{s}m_{ij}{}^{(4)}$$

② 画面モードがモノクロの場合

・画面イメージ

①と同じ

・メモリ上の格納域イメージ（バイト表現）

| オフセット | | | | |
|---|---|---|---|---|
| +0 | X方向ドット数m | | | |
| +2 | Y方向ドット数n | | | |
| +(4) | $M_{1,1}$，$M_{1,2}$， | ………………… | ，$M_{1,k}$ |
| +(4+k) | $M_{2,1}$，$M_{2,2}$， | ………………… | ，$M_{2,k}$ |
| +(4+2k) | $M_{3,1}$，$M_{3,2}$， | ………………… | ，$M_{3,k}$ |
| ⋮ | ⋮ | | |
| +(4+(n−1)k) | $M_{n,1}$，$M_{n,2}$， | ………………… | ，$M_{nk}$ |

k＝(X2−X1+8)¥8

・格納域イメージのバイトと対応ビットとの関係

$$M_{ij} = ({}^{7}m_{ij},\ {}^{6}m_{ij},\ {}^{5}m_{ij},\ {}^{4}m_{ij},\ {}^{3}m_{ij},\ {}^{2}m_{ij},\ {}^{1}m_{ij},\ {}^{0}m_{ij})$$

${}^{s}m_{ij}=0$または$1$（$s=0\sim7,\ i=1\sim n,\ j=1\sim k$）

・画面のイメージのドットと格納域バイトとの対応

$V_{i1}$, $V_{i2}$, ……, $V_{i7}$, $V_{i8}$, $V_{i9}$, $V_{i10}$, ……, ………

$^7m_{i1}$, $^6m_{i1}$, ……, $^1m_{i1}$, $^0m_{i1}$, $^7m_{i2}$, $^6m_{i2}$, ……,

・各ドットの白黒表示,ドットに対応するビットが1のとき白，0のとき黒.

## 4.14　描画情報を戻す(GPUT1)

### (1) 機能

指定格納域内の描画情報を，指定の領域上に戻す.

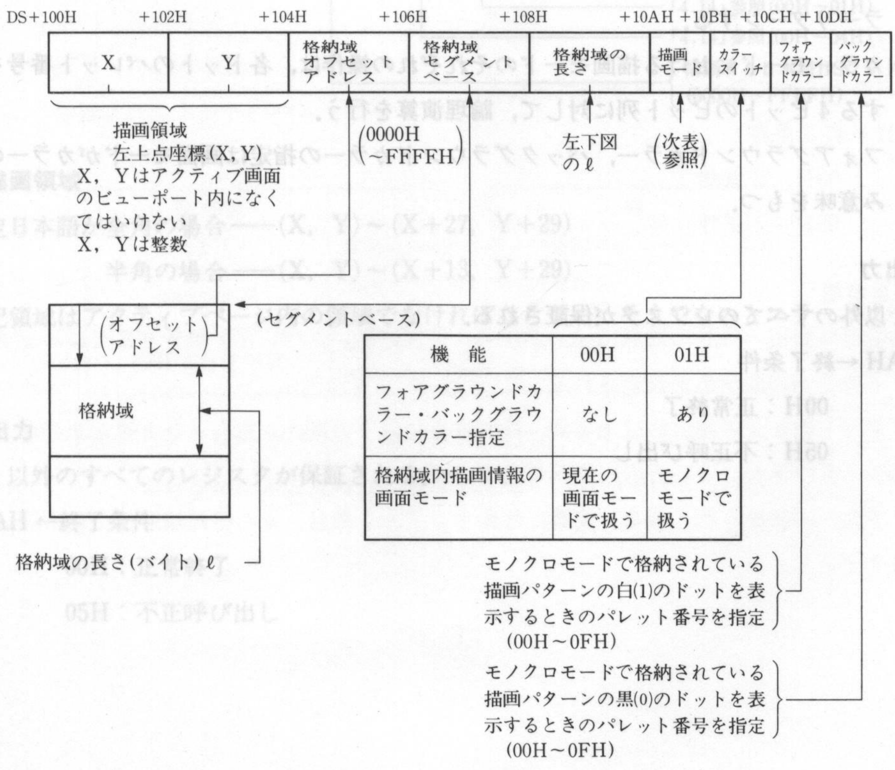
### (2) 入力条件

・内部割り込みコード← 1DH

・AH ← 0CH

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




| 機　能 | 00H | 01H |
|---|---|---|
| フォアグラウンドカラー・バックグラウンドカラー指定 | なし | あり |
| 格納域内描画情報の画面モード | 現在の画面モードで扱う | モノクロモードで扱う |

〈描画モード〉

指定領域上の現在の描画パターンを $A_0$,

格納域の描画パターンを B,

格納域の描画パターンによって操作した指定領域上の描画パターンを $A_N$ としたとき，$A_0$ に対して B で OP 操作を行った結果が $A_N$ とすると，

$$A_0 \ \mathrm{OP} \ B \rightarrow A_N$$

描画モード(00H～04H)は次のような操作を表わす.

| 描画モード | 操　　作 | 説　　明 |
|---|---|---|
| 00H | $B \rightarrow A_N$ | |
| 01H | $\overline{B} \rightarrow A_N$ | |
| 02H | $A_0 + B \rightarrow A_N$ | 論理和 |
| 03H | $A_0 \times B \rightarrow A_N$ | 論理積 |
| 04H | $A_0 - B \rightarrow A_N$ | 排他的論理和 |

・格納域(描画情報)の形式は，「4.13　GGET」を参照のこと

注：描画領域左上点(X，Y)，右下点(X+X 方向ドット数－1，Y+Y 方向ドット数－1)はアクティブページの描画領域内になければならない．そうでないと，処理は行われずエラーリターンする．

カラーモードにおける描画モードのそれぞれの操作は，各ドットのパレット番号を表現する4ビットのビット列に対して，論理演算を行う．

フォアグラウンドカラー，バックグラウンドカラーの指定は画面モードがカラーの時のみ意味をもつ．

### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH←終了条件

00H：正常終了

05H：不正呼び出し

# 4.15　日本語の描画(GPUT2)

## (1) 機能

指定の日本語(JIS コード)を，指定の領域上に描画する．

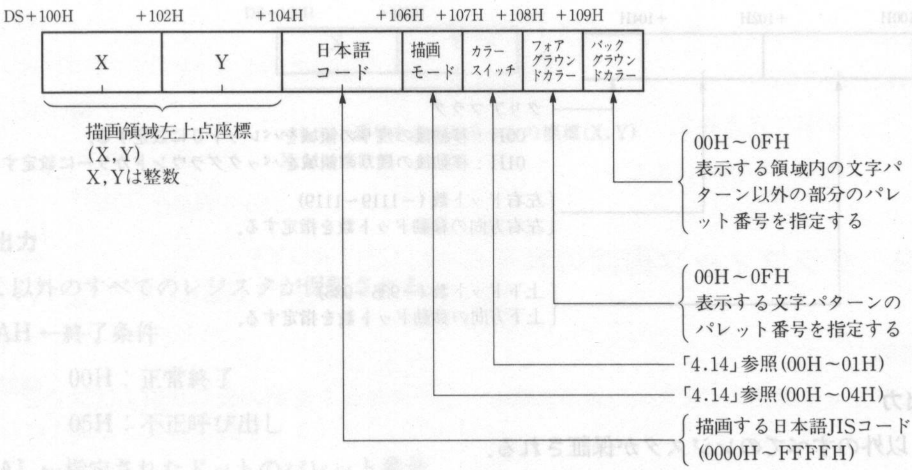
## (2) 入力

・内部割り込みコード← 1DH

・AH ← 0DH

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




## (3) 描画領域

指定日本語が全角の場合――(X,　Y)～(X+27,　Y+29)

半角の場合――(X,　Y)～(X+13,　Y+29)

上記領域はアクティブページ内の領域でなければならない．そうでないとエラーリターンする．

## (4) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

## 4.16　描画画面の移動(GROLL)

### (1) 機能

アクティブ画面全体の描画情報を，指定ドット数分上下または左右方向へ移動する．

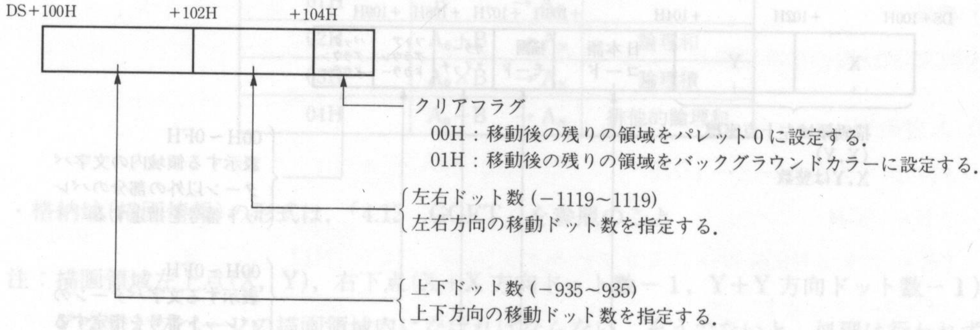
### (2) 入力

・内部割り込みコード← 1DH

・AH ← 0EH

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

　　00H：正常終了

　　05H：不正呼び出し

### (4) 注意事項

① 描画情報移動後の残りの領域には，クリアフラグに従い，パレット番号0またはバックグラウンドカラーの表示色が設定される．

② 上下ドット数が正の場合には上方向，負の場合には下方向へ，左右ドット数が正の場合には左方向，負の場合には右方向へ描画情報を移動する．

③ 左右方向への移動を行う場合，実際に移動するドット数は，その絶対値以下で最も近い8の倍数分である．

## 4.17　ドットに対応するパレット番号の取得(GPOINT2)

### (1) 機能

指定座標のドットのパレット番号を取得する．

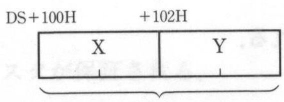
### (2) 入力

・内部割り込みコード← 1DH

・AH ← 0FH

・DS ←グラフ BIOS ワークエリアのセグメントベース(GINIT と同じ値)

・パラメータリスト




パレット番号を求めるドットの座標(X, Y)
X, Y は整数値

### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

　　00H：正常終了
　　05H：不正呼び出し

・AL ←指定されたドットのパレット番号

| AL の値 | 内　容 |
|---|---|
| FFH | 指定座標がアクティブ画面のビューポート以外 |
| 00H～0FH | 画面モードがカラーの場合：<br>指定座標のパレット番号を示す． |
| 00H または 01H | 画面モードがモノクロの場合：<br>00H－黒，01H－白 |

注：モノクロ時のパレット番号には，設定された表示が与えられているが，未設定時は，0 が 000H，1 が FFFH になる．

## 4.18　表示領域の設定(GSCROLL)

### (1) 機能

表示対象とする描画メモリ領域を設定する.

### (2) 入力

・内部割り込みコード←1DH

・AH←10H

・DS←グラフBIOSワークエリア(ユーザーUCW)のセグメントベース(GINITと同じ値)

・DS：100～DS：101←表示開始Y座標(0～186)

### (3) 出力

AX以外のすべてのレジスタが保証される.

・AH←終了条件

00H：正常終了

05H：不正呼び出し

## 4.19　グラフィック画面の表示開始(GSTART)

### (1) 機能

グラフィック画面のCRTへの表示を開始する．テキスト画面の表示状態とは独立して機能する.

### (2) 入力

・内部割り込みコード←1DH

・AH←11H

・DS←グラフBIOSワークエリア(ユーザーUCW)のセグメントベース(GINITと同じ値)

### (3) 出力

AX以外のすべてのレジスタが保証される.

・AH←終了条件

00H：正常終了

05H：不正呼び出し

## 4.20　グラフィック画面の表示停止(GSTOP)

### (1) 機能

グラフィック画面の CRT への表示を停止する．テキスト画面の表示状態とは独立して機能する．

### (2) 入力

・内部割り込みコード← 1DH

・AH ← 12H

・DS ←グラフ BIOS ワークエリア(ユーザー UCW)のセグメントベース(GINIT と同じ値)

### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

## 4.21　グラフ BIOS の終了(GTERM)

### (1) 機能

グラフ BIOS の使用を終了する．

### (2) 入力

・内部割り込みコード← 1DH

・AH ← 13H

・DS ←グラフ BIOS ワークエリア(ユーザー UCW)のセグメントベース(GINIT と同じ値)

### (3) 出力

AX 以外のすべてのレジスタが保証される．

・AH ←終了条件

00H：正常終了

05H：不正呼び出し

● **サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;###########################################################
;#                                                         #
;#      G r a p h   B I O Sの初期化をする                  #
;#                                                         #
;#              表示モード：カラー                          #
;#              表示スイッチ：GRAPHIC 表示なし              #
;#              ACTIVE PAGE : PAGE Ø                       #
;#              DISPLAY PAGE: PAGE 1                       #
;#                                                         #
;###########################################################
GRAPH   GROUP   CODE,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS

        END     START
```

```
;###########################################################
;#                                                         #
;#      グラフィック画面のモードを設定する                  #
;#                                                         #
;#              表示モード：カラー                          #
;#              表示スイッチ：GRAPHIC 表示あり・普通描画    #
;#              ACTIVE PAGE : PAGE Ø                       #
;#              DISPLAY PAGE: PAGE Ø                       #
;#                                                         #
;###########################################################
GRAPH   GROUP   CODE,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø          ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø          ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø          ;アクティブページ:Ø
        MOV     BYTE PTR DS:[1Ø3H],1          ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS

        END     START
```

```
;##############################################################
;#                                                            #
;#      グラフィック画面の描画領域を設定し，塗りつぶす        #
;#                                                            #
;##############################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER              ;作業領域セグメント
        MOV     DS,AX
;                                          ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                          ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø           ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø           ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø           ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1           ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                          ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],5Ø          ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],1ØØ         ;  (5Ø,1ØØ)-(98Ø,68Ø)
        MOV     WORD PTR DS:[1Ø4H],98Ø
        MOV     WORD PTR DS:[1Ø6H],68Ø
        MOV     BYTE PTR DS:[1Ø8H],Ø9H         ;描画領域色:Ø9h
        MOV     BYTE PTR DS:[1Ø9H],ØEH         ;外枠色:ØEh
        MOV     AH,2
        INT     1DH
;                                          ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                     ;キー入力待ち
        INT     21H
;                                          ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                  ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø           ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø           ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH          ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH        ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                     ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      Ødh,Øah,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)                 ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                 ;パラメータリスト
GWORK   DB      512 DUP(?)                 ;GRAPH BIOS 作業領域
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                                #
;#        前景色と背景色を指定する                                #
;#                                                                #
;##################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0    ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0    ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0    ;アクティブページ:0
        MOV     BYTE PTR DS:[103H],1    ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
        CALL    PAINT                   ;画面消去/四辺形を塗る
;
        MOV     AX,SEG DATA             ;メッセージを表示
        MOV     DS,AX
        MOV     DX,OFFSET MSG1
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                  ;キー入力待ち
        INT     21H
;                                       ;カラーの変更
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
        MOV     BYTE PTR DS:[100H],0FH  ;背景色:0FH
        MOV     BYTE PTR DS:[101H],01H  ;前景色:01H
        MOV     AH,3
        INT     1DH
        CALL    PAINT                   ;画面消去/四辺形を塗る
;
        MOV     AX,SEG DATA             ;メッセージを表示
        MOV     DS,AX
        MOV     DX,OFFSET MSG2
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                  ;キー入力待ち
        INT     21H
;                                       ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0    ;描画領域:
        MOV     WORD PTR DS:[102H],0    ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H   ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
;                                       ;背景色でクリア
PAINT:  MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
        MOV     AH,5
        INT     1DH
;                                       ;四辺形を描き,塗りつぶす
        MOV     WORD PTR DS:[100H],600  ;描画範囲:
        MOV     WORD PTR DS:[102H],100  ;  (600,100)-(800,400)
        MOV     WORD PTR DS:[104H],800
        MOV     WORD PTR DS:[106H],400
        MOV     BYTE PTR DS:[108H],0FFH ;描画色:前景色
        MOV     BYTE PTR DS:[109H],02H  ;四辺形/塗りつぶし
        MOV     BYTE PTR DS:[10AH],00H  ;スイッチ指定なし
        MOV     AH,7
        INT     1DH
        RET
CODE    ENDS
;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG1    DB      '何かキーを押すと,色が変わります.',0dh,0ah,'$'
MSG2    DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0dh,0ah,'$'
DATA    ENDS
;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS
;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS
        END     START
```

```
;#################################################################
;#                                                               #
;#       パレット番号を割り振る                                    #
;#                                                               #
;#################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø           ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø           ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø           ;アクティブページ :Ø
        MOV     BYTE PTR DS:[1Ø3H],1           ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø           ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø           ;  (Ø,Ø)-(1Ø24,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],Ø8H         ;描画領域色:Ø8H
        MOV     BYTE PTR DS:[1Ø9H],ØFFH        ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG1
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                  ;キー入力待ち
        INT     21H
;                                       ;パレット番号の割り振り
        MOV     AX,SEG BUFFER                  ;作業領域セグメント
        MOV     DS,AX
        MOV     BYTE PTR DS:[1ØØH],Ø8H         ;パレット番号:Ø8H
        MOV     WORD PTR DS:[1Ø1H],Ø1A5H       ;表示色:1A5H
        MOV     AH,4
        INT     1DH
;                                       ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG2
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                  ;キー入力待ち
        INT     21H
;                                       ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                  ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø           ;描画領域は:
        MOV     WORD PTR DS:[1Ø2H],Ø           ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH          ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH        ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG1    DB      '何かキーを押すと,パレットの割当て色が変わります.'
        DB      Ødh,Øah,'$'
MSG2    DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      Ødh,Øah,'$'
DATA    ENDS
;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS
;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS
        END     START
```

```
;#################################################################
;#                                                               #
;#      描画領域内を背景色で塗りつぶす                           #
;#                                                               #
;#################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[100H],0            ;描画領域
        MOV     WORD PTR DS:[102H],0            ;   (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H          ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠の指定なし
        MOV     AH,2
        INT     1DH
;                                       ;描画カラーを設定
        MOV     BYTE PTR DS:[100H],01H          ;背景色:01H
        MOV     BYTE PTR DS:[101H],0EH          ;前景色:0EH
        MOV     AH,3
        INT     1DH
;                                       ;画面クリア
        MOV     AX,SEG BUFFER                   ;作業領域セグメント
        MOV     DS,AX
        MOV     AH,5
        INT     1DH
;                                       ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                  ;キー入力待ち
        INT     21H
;                                       ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                   ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0            ;描画領域は:
        MOV     WORD PTR DS:[102H],0            ;   (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H          ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                          ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0dh,0ah,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#     描画領域内の座標に,指定するパレット番号の点を表示する         #
;##################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[100H],0            ;描画領域
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H           ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;10回繰返し
        MOV     CH,10                   ;{
        MOV     BX,200
;                                               ;点を描く
LOOP:   MOV     WORD PTR DS:[100H],640          ;描画位置:
        MOV     WORD PTR DS:[102H],BX           ;  (640,200+n)
        MOV     BYTE PTR DS:[104H],0EH          ;描画色:0EH
        MOV     AH,6
        MOV     AL,1
        INT     1DH
        ADD     BX,10                   ;}
        DEC     CH
        JNE     LOOP
;                                       ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                  ;キー入力待ち
        INT     21H
;                                       ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                   ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H           ;描画色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      'X座標 640, Y座標 200 に点が描かれています.'
        DB      0dh,0ah
        DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0dh,0ah,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84H DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################
;#      直線・四辺形・塗りつぶした四辺形を描く       #
;#                                                #
;##################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:  MOV     AX,SEG BUFFER            ;作業領域セグメントを設定
        MOV     DS,AX
;                                        ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                        ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø     ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø     ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø     ;アクティブページ:Ø
        MOV     BYTE PTR DS:[1Ø3H],1     ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                        ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø     ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø     ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH    ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH  ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                        ;描画カラーを設定
        MOV     BYTE PTR DS:[1ØØH],ØØH   ;背景色:ØØH
        MOV     BYTE PTR DS:[1Ø1H],ØEH   ;前景色:ØEH
        MOV     AH,3
        INT     1DH
;                                        ;直線を描く
LINE:   MOV     WORD PTR DS:[1ØØH],2Ø    ;描画範囲:
        MOV     WORD PTR DS:[1Ø2H],2Ø    ;  (2Ø,2Ø)-(1ØØØ,1ØØ)
        MOV     WORD PTR DS:[1Ø4H],1ØØØ
        MOV     WORD PTR DS:[1Ø6H],1ØØ
        MOV     BYTE PTR DS:[1Ø8H],ØFFH  ;描画色:前景色
        MOV     BYTE PTR DS:[1Ø9H],ØØH   ;直線を描く
        MOV     BYTE PTR DS:[1ØAH],Ø1H   ;ラインスタイルあり
        MOV     WORD PTR DS:[1ØBH],ØFFØØH ;ラインスタイル
        MOV     AH,7
        INT     1DH
;                                        ;四辺形を描く
BOX:    MOV     WORD PTR DS:[1ØØH],1ØØ   ;描画範囲:
        MOV     WORD PTR DS:[1Ø2H],1ØØ   ;  (1ØØ,1ØØ)-(8ØØ,3ØØ)
        MOV     WORD PTR DS:[1Ø4H],8ØØ
        MOV     WORD PTR DS:[1Ø6H],3ØØ
        MOV     BYTE PTR DS:[1Ø8H],Ø9H   ;描画色:Ø9H
        MOV     BYTE PTR DS:[1Ø9H],Ø1H   ;四辺形を描く
        MOV     BYTE PTR DS:[1ØAH],ØØH   ;スイッチ指定なし
        MOV     AH,7
        INT     1DH
;                                        ;四辺形を描き,塗りつぶす
BOXFLL: MOV     WORD PTR DS:[1ØØH],5ØØ   ;描画範囲:
        MOV     WORD PTR DS:[1Ø2H],4ØØ   ;  (5ØØ,4ØØ)-(9ØØ,7ØØ)
        MOV     WORD PTR DS:[1Ø4H],9ØØ
        MOV     WORD PTR DS:[1Ø6H],7ØØ
        MOV     BYTE PTR DS:[1Ø8H],ØBH   ;描画色:ØBH
        MOV     BYTE PTR DS:[1Ø9H],Ø2H   ;四辺形/塗りつぶし
        MOV     BYTE PTR DS:[1ØAH],ØØH   ;スイッチ指定なし
        MOV     AH,7
        INT     1DH
;                                        ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                   ;キー入力待ち
        INT     21H
;                                        ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER            ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø     ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø     ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH    ;描画色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH  ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:   MOV     AH,4CH                   ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      Ødh,Øah,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
```

```
;######### GRAPH BIOS 使用作業領域
BUFFER   SEGMENT PARA PUBLIC 'GRAPH'
GBIOS    DB    256 DUP(?)         ;GRAPH BIOS UCW
PARALST  DB    128 DUP(?)         ;パラメータリスト
GWORK    DB    512 DUP(?)         ;GRAPH BIOS 作業領域
BUFFER   ENDS

;######### スタック領域
STACK    SEGMENT PARA STACK 'STACK'
         DW    84H DUP (?)
STACK    ENDS

         END   START
```

```
;################################################################
;#                                                              #
;#       円(楕円)・扇形を描く                                    #
;#                                                              #
;################################################################
GRAPH    GROUP    CODE,DATA,INTRPT,BUFFER
CODE     SEGMENT  PARA PUBLIC 'GRAPH'
         ASSUME   CS:CODE,DS:BUFFER,SS:STACK,ES:CODE
START:
         MOV      AX,SEG BUFFER             ;作業領域セグメントを設定
         MOV      DS,AX
;                                           ;GRAPH BIOS の初期設定
         MOV      WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
         MOV      WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
         MOV      AH,Ø
         INT      1DH
;                                           ;グラフ画面のモードを設定
         MOV      BYTE PTR DS:[1ØØH],Ø          ;表示モード:カラー
         MOV      BYTE PTR DS:[1Ø1H],Ø          ;表示SW:表示あり,普通
         MOV      BYTE PTR DS:[1Ø2H],Ø          ;アクティブページ  :Ø
         MOV      BYTE PTR DS:[1Ø3H],1          ;ディスプレイページ:1
         MOV      AH,1
         INT      1DH
;                                           ;描画領域を設定し,塗る
         MOV      WORD PTR DS:[1ØØH],Ø          ;描画領域:
         MOV      WORD PTR DS:[1Ø2H],Ø          ;  (Ø,Ø)-(1119,749)
         MOV      WORD PTR DS:[1Ø4H],1119
         MOV      WORD PTR DS:[1Ø6H],749
         MOV      BYTE PTR DS:[1Ø8H],ØH         ;描画領域色:ØØH
         MOV      BYTE PTR DS:[1Ø9H],ØFFH       ;外枠は描かない
         MOV      AH,2
         INT      1DH
;                                           ;円を描く
EN:      MOV      WORD PTR DS:[1ØØH],32Ø        ;中心座標:(32Ø,2ØØ)
         MOV      WORD PTR DS:[1Ø2H],2ØØ
         MOV      WORD PTR DS:[1Ø4H],8Ø         ;X方向半径:8Ø
         MOV      WORD PTR DS:[1Ø6H],8Ø         ;Y方向半径:8Ø
         MOV      BYTE PTR DS:[1Ø8H],ØAH        ;表示色:ØAH
         MOV      BYTE PTR DS:[1Ø9H],Ø          ;フラグ指定なし
         MOV      AH,8
         INT      1DH
;                                           ;半円を描く
HANEN:   MOV      WORD PTR DS:[1ØØH],512        ;中心座標:(512,336)
         MOV      WORD PTR DS:[1Ø2H],336
         MOV      WORD PTR DS:[1Ø4H],8Ø         ;X方向半径:8Ø
         MOV      WORD PTR DS:[1Ø6H],8Ø         ;Y方向半径:8Ø
         MOV      BYTE PTR DS:[1Ø8H],ØDH        ;表示色:ØDH
         MOV      BYTE PTR DS:[1Ø9H],Ø5H        ;開始・終了座標あり
                                                ;終了線あり
         MOV      WORD PTR DS:[1ØAH],592        ;開始X座標:592
         MOV      WORD PTR DS:[1ØCH],336        ;開始Y座標:336
         MOV      WORD PTR DS:[1ØEH],432        ;終了X座標:432
         MOV      WORD PTR DS:[11ØH],336        ;終了Y座標:336
```

```
        MOV     AH,8
        INT     1DH
;
OVAL:   MOV     WORD PTR DS:[100H],368       ;楕円を描く
        MOV     WORD PTR DS:[102H],560       ;中心座標:(368,560)
        MOV     WORD PTR DS:[104H],144       ;X方向座標:144
        MOV     WORD PTR DS:[106H],80        ;Y方向座標:80
        MOV     BYTE PTR DS:[108H],0CH       ;表示色:0CH
        MOV     BYTE PTR DS:[109H],25H       ;開始・終了座標あり
                                             ;塗りつぶしあり
        MOV     WORD PTR DS:[10AH],512       ;開始X座標:512
        MOV     WORD PTR DS:[10CH],560       ;開始Y座標:560
        MOV     WORD PTR DS:[10EH],512       ;終了X座標:512
        MOV     WORD PTR DS:[110H],560       ;終了Y座標:560
        MOV     BYTE PTR DS:[112H],0BH       ;パレット番号指定
        MOV     AH,8
        INT     1DH
;                                            ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                       ;キー入力待ち
        INT     21H
;                                            ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0         ;描画領域:
        MOV     WORD PTR DS:[102H],0         ;   (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H        ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH      ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                       ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP(?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                               #
;#  座標点と境界色で指定する領域を指定色で塗りつぶす              #
;#                                                               #
;##################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK,ES:CODE
START:  MOV     AX,SEG BUFFER          ;作業領域セグメントを設定
        MOV     DS,AX
;                                      ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                      ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø       ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø       ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø       ;アクティブページ   :Ø
        MOV     BYTE PTR DS:[1Ø3H],1       ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                      ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø       ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø       ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH      ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH    ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                      ;四辺形を描く
BOX:    MOV     WORD PTR DS:[1ØØH],64Ø     ;描画範囲:
        MOV     WORD PTR DS:[1Ø2H],2ØØ     ; (64Ø,2ØØ)-(768,328)
        MOV     WORD PTR DS:[1Ø4H],768
        MOV     WORD PTR DS:[1Ø6H],328
        MOV     BYTE PTR DS:[1Ø8H],ØFH     ;描画色:OFH
        MOV     BYTE PTR DS:[1Ø9H],Ø1H     ;四辺形を描く
        MOV     BYTE PTR DS:[1ØAH],ØØH     ;スイッチの指定なし
        MOV     AH,7
        INT     1DH
;                                      ;指定した線分の中を塗る
PAINT:  MOV     WORD PTR DS:[1ØØH],7ØØ     ;開始X座標:7ØØ
        MOV     WORD PTR DS:[1Ø2H],24Ø     ;開始Y座標:24Ø
        MOV     BYTE PTR DS:[1Ø4H],ØAH     ;領域色:ØAH
        MOV     BYTE PTR DS:[1Ø5H],ØFH     ;境界色:ØFH
        MOV     AX,OFFSET WORKST           ;作業領域先頭オフセット
        MOV     WORD PTR DS:[1Ø6H],AX
        MOV     AX,OFFSET ENDWRK           ;作業領域最終オフセット
        MOV     WORD PTR DS:[1Ø8H],AX
        MOV     AH,9
        INT     1DH
;                                      ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                 ;キー入力待ち
        INT     21H
;                                      ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER          ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø       ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø       ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH      ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH    ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                 ;プログラムを終了
        INT     21H
CODE    ENDS
;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      ØDH,ØAH,'$'
DATA    ENDS
;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)             ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)             ;パラメータリスト
GWORK   DB      512 DUP(?)             ;GRAPH BIOS 作業領域
WORKST  DB      1ØØH DUP(?)            ;GPAINT1 作業領域
ENDWRK  LABEL   BYTE
BUFFER  ENDS
;######### スタック領域
STACK   SEGMENT PARA STACK'STACK'
        DW      84 DUP (?)
STACK   ENDS
        END     START
```

```
;###############################################################
;#                                                             #
;#   座標点と境界色で指定する領域をタイルパターンで塗りつぶす  #
;#                                                             #
;###############################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK,ES:CODE
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H           ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;四辺形を描く
BOX:    MOV     WORD PTR DS:[100H],500          ;描画範囲:
        MOV     WORD PTR DS:[102H],200          ;  (500,200)-(800,500)
        MOV     WORD PTR DS:[104H],800
        MOV     WORD PTR DS:[106H],500
        MOV     BYTE PTR DS:[108H],0FH          ;描画色:0FH
        MOV     BYTE PTR DS:[109H],01H          ;四辺形を描く
        MOV     BYTE PTR DS:[10AH],00H          ;スイッチの指定なし
        MOV     AH,7
        INT     1DH
;                                       ;指定した線分の中を塗る
PAINT:  MOV     WORD PTR DS:[100H],700          ;開始X座標:700
        MOV     WORD PTR DS:[102H],240          ;開始Y座標:240
        MOV     BYTE PTR DS:[104H],4            ;タイルパターン長
        MOV     WORD PTR DS:[105H],OFFSET TIL   ;パターンオフセット
        MOV     AX,SEG TIL
        MOV     WORD PTR DS:[107H],AX           ;パターンセグメント
        MOV     BYTE PTR DS:[109H],0FH          ;境界色:0FH
        MOV     AX,OFFSET WORKST                ;作業域先頭オフセット
        MOV     WORD PTR DS:[10AH],AX
        MOV     AX,OFFSET ENDWRK                ;作業域最終オフセット
        MOV     WORD PTR DS:[10CH],AX
        MOV     AH,0AH
        INT     1DH
;                                               ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                          ;キー入力待ち
        INT     21H
                                                ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                   ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H           ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;
EXIT:   MOV     AH,4CH                          ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0DH,0AH,'$'
TIL     DB      55H
        DB      0AAH
        DB      55H
        DB      0AAH
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)                      ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                      ;パラメータリスト
GWORK   DB      512 DUP(?)                      ;GRAPH BIOS 作業領域
WORKST  DB      100H DUP(?)                     ;GPAINT1 作業領域
```

```
;###################################################################
;#                                                                 #
;#        ＶＲＡＭ上の描画情報をメモリに取り込む                         #
;#                                                                 #
;###################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK,ES:CODE
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø          ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø          ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø          ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1          ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø          ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø          ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH        ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH       ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;四辺形を描く
BOX:    MOV     WORD PTR DS:[1ØØH],64Ø        ;描画範囲:
        MOV     WORD PTR DS:[1Ø2H],2ØØ        ;  (64Ø,2ØØ)-(768,328)
        MOV     WORD PTR DS:[1Ø4H],768
        MOV     WORD PTR DS:[1Ø6H],328
        MOV     BYTE PTR DS:[1Ø8H],ØEH        ;描画色:OEH
        MOV     BYTE PTR DS:[1Ø9H],Ø2H        ;四辺形を描/塗る
        MOV     BYTE PTR DS:[1ØAH],ØØH        ;ラインスタイルなし
        MOV     AH,7
        INT     1DH
;                                       ;描画情報を取り込む
GGET:   MOV     WORD PTR DS:[1ØØH],64Ø        ;格納範囲:
        MOV     WORD PTR DS:[1Ø2H],2ØØ        ;  (64Ø,2ØØ)-(688,248)
        MOV     WORD PTR DS:[1Ø4H],7ØØ
        MOV     WORD PTR DS:[1Ø6H],26Ø
        MOV     WORD PTR DS:[1Ø8H],OFFSET GET ;格納域のオフセット
        MOV     AX,SEG GET                    ;格納域セグメント
        MOV     WORD PTR DS:[1ØAH],AX
        MOV     WORD PTR DS:[1ØCH],14ØØ       ;格納域サイズ
        MOV     AH,ØBH
        INT     1DH
```

```
ENDWRK  LABEL   BYTE
BUFFER  ENDS
;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS
        END     START
```

```
;###############################################################
;#                                                             #
;#        メモリの格納域に取り込んだ描画情報をＶＲＡＭ上に書く        #
;#                                                             #
;###############################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:  MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,ØØH
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø    ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø    ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø    ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1    ;ディスプレイページ:1
        MOV     AH,Ø1H
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø    ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø    ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH  ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH ;外枠は描かない
        MOV     AH,Ø2H
        INT     1DH
;                                       ;四辺形を描く
BOX:    MOV     WORD PTR DS:[1ØØH],ØH   ;描画位置:
        MOV     WORD PTR DS:[1Ø2H],ØH   ;(64Ø,2ØØ)-(768,328)
        MOV     WORD PTR DS:[1Ø4H],8ØH
        MOV     WORD PTR DS:[1Ø6H],8ØH
        MOV     BYTE PTR DS:[1Ø8H],ØFH  ;描画色:OFH
        MOV     BYTE PTR DS:[1Ø9H],Ø2H  ;四辺形を描く/塗る
        MOV     BYTE PTR DS:[1ØAH],ØØH  ;パレット番号２なし
        MOV     AH,Ø7H
        INT     1DH
;                                       ;VRAMの描画情報を格納
GGET:   MOV     WORD PTR DS:[1ØØH],Ø    ;取り込む範囲:
        MOV     WORD PTR DS:[1Ø2H],Ø    ; (Ø,Ø)-(128,128)
        MOV     WORD PTR DS:[1Ø4H],128
        MOV     WORD PTR DS:[1Ø6H],128
        MOV     WORD PTR DS:[1Ø8H],OFFSET GET ;格納域のオフセット
        MOV     AX,SEG GET              ;格納域のセグメント
        MOV     WORD PTR DS:[1ØAH],AX
        MOV     WORD PTR DS:[1ØCH],23ØØH ;格納域のサイズ
        MOV     AH,ØBH
        INT     1DH
;                                       ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                  ;キー入力待ち
        INT     21H
;                                       ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø    ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø    ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH   ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      ØDH,ØAH,'$'
GET     DB      14ØØ DUP(?)             ;格納域
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
WORKST  DB      1ØØH DUP(?)             ;GPAINT1 作業領域
ENDWRK  LABEL   BYTE
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS

        END     START
```

```
;                                    ;格納した内容を書き出す
GPUT:   MOV     WORD PTR DS:[100H],300       ;書き出し位置:
        MOV     WORD PTR DS:[102H],200       ;  (300,200)
        MOV     WORD PTR DS:[104H],OFFSET GET  ;格納域のオフセット
        MOV     AX,SEG GET                   ;格納域のセグメント
        MOV     WORD PTR DS:[106H],AX
        MOV     WORD PTR DS:[108H],2300H     ;格納域のサイズ
        MOV     BYTE PTR DS:[10AH],0
        MOV     BYTE PTR DS:[10BH],0         ;カラースイッチなし
        MOV     AH,0CH
        INT     1DH
;                                            ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
;                                            ;キー入力待ち
        MOV     AH,01H
        INT     21H
;                                            ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0         ;描画領域:
        MOV     WORD PTR DS:[102H],0         ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H        ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH      ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                       ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0DH,0AH,'$'
;
GET     DB      2300H DUP(?)
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)                   ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                   ;パラメータリスト
GWORK   DB      512 DUP(?)                   ;GRAPH BIOS 作業領域
WORKST  DB      100H DUP(?)                  ;GPAINT1 作業領域
ENDWRK  LABEL   BYTE
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS

        END     START
```

```
;####################################################################
;#                                                                 #
;#      漢字をグラフィック画面に表示する                              #
;#                                                                 #
;####################################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER               ;作業領域セグメント
        MOV     DS,AX
;                                           ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                           ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                           ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],08H          ;描画領域色:08H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
KANJI:                                      ;漢字を表示
        MOV     WORD PTR DS:[100H],600          ;表示位置:
        MOV     WORD PTR DS:[102H],200          ;  (600,200)
        MOV     WORD PTR DS:[104H],3026H        ;JIS コード:3026H(愛)
        MOV     BYTE PTR DS:[106H],0            ;重ね書き
        MOV     BYTE PTR DS:[107H],01H          ;色指定あり
        MOV     BYTE PTR DS:[108H],0FH          ;前景色:0FH
        MOV     BYTE PTR DS:[109H],0AH          ;背景色:0AH
        MOV     AH,0DH
        INT     1DH
;                                           ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                      ;キー入力待ち
        INT     21H
;                                           ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER               ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0H           ;描画領域色:00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                          ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      0DH,0AH,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)                  ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                  ;パラメータリスト
GWORK   DB      512 DUP(?)                  ;GRAPH BIOS 作業領域
WORKST  DB      100H DUP(?)                 ;GPAINT1 作業領域
ENDWRK  LABEL   BYTE
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS
        END     START
```

```
;##################################################
;#                                                #
;#     アクティブ画面全体を,上下左右にスクロールする   #
;#                                                #
;##################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø        ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø        ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø        ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1        ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø        ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø        ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØFH      ;描画領域色:ØFH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH     ;外枠は描かない
        MOV     AH,2
        INT     1DH
BOXFUL:                                 ;塗りつぶした四辺形を描く
        MOV     WORD PTR DS:[1ØØH],4ØØ      ;描画位置:
        MOV     WORD PTR DS:[1Ø2H],4ØØ      ;(4ØØ,4ØØ)-(5ØØ,55Ø)
        MOV     WORD PTR DS:[1Ø4H],5ØØ
        MOV     WORD PTR DS:[1Ø6H],55Ø
        MOV     BYTE PTR DS:[1Ø8H],ØEH      ;描画色:ØEH
        MOV     BYTE PTR DS:[1Ø9H],Ø2H      ;塗りつぶす
        MOV     BYTE PTR DS:[1ØAH],ØØH      ;スイッチ指定なし
        MOV     AH,7
        INT     1DH
;
        MOV     CX,16
;                                       ;グラフ画面を右上へロール
GROLL:  MOV     WORD PTR DS:[1ØØH],2Ø       ;上へ2Øドットロール
        MOV     WORD PTR DS:[1Ø2H],16       ;左へ16ドットロール
        MOV     BYTE PTR DS:[1Ø4H],ØØH      ;残り領域の色:ØØH
        MOV     AH,ØEH
        INT     1DH
        LOOP    GROLL
;                                       ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                      ;キー入力待ち
        INT     21H
;                                           ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER               ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø        ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø        ;  (Ø,Ø)-(1119,935)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],935
        MOV     BYTE PTR DS:[1Ø8H],ØH       ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH     ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                      ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)                  ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                  ;パラメータリスト
GWORK   DB      512 DUP(?)                  ;GRAPH BIOS 作業領域
WORKST  DB      1ØØH DUP(?)                 ;GPAINT1 作業領域
ENDWRK  LABEL   BYTE
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS
        END     START
```

```
;############################################################
;#                                                          #
;#      指定した座標の色（パレット番号）を調べる            #
;#                                                          #
;############################################################
GRAPH   GROUP   CODE,DATA,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MQV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0        ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0        ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0        ;アクティブページ:0
        MOV     BYTE PTR DS:[103H],1        ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定する
        MOV     WORD PTR DS:[100H],0        ;描画領域:
        MOV     WORD PTR DS:[102H],0        ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],0FFH     ;描画領域を塗らない
        MOV     BYTE PTR DS:[109H],0FFH     ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;描画カラーを設定
        MOV     BYTE PTR DS:[100H],00H      ;背景色:00H
        MOV     BYTE PTR DS:[101H],0FH      ;前景色:0FH
        MOV     AH,3
        INT     1DH
;                                       ;画面クリア
        MOV     AH,5
        INT     1DH
;                                       ;点を描く
        MOV     WORD PTR DS:[100H],640      ;描画位置:
        MOV     WORD PTR DS:[102H],400      ;  (640,400)
        MOV     BYTE PTR DS:[104H],0FH      ;描画色:0FH
        MOV     AL,1
        MOV     AH,6
        INT     1DH
POINT:                                  ;指定座標の色を調べる
        MOV     WORD PTR DS:[100H],640      ;指定座標:
        MOV     WORD PTR DS:[102H],400      ;  (640,400)
        MOV     AH,0FH
        INT     1DH
;
        MOV     BX,SEG DATA             ;データセグメント設定
        MOV     DS,BX
        CMP     AL,0AH                  ;パレット番号をセット
        JNB     AAAA
        ADD     AL,30H                      ;パレット番号 1-9
        JMP     DISP
;
AAAA:   SUB     AL,0AH                      ;パレット番号 A-F
        ADD     AL,41H
DISP:   MOV     PALET,AL
        MOV     AH,09H                  ;メッセージを表示
        MOV     DX,OFFSET MSG1
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG1    DB      0DH,0AH,'座標[640,400]のパレット番号は『0'
PALET   DB      '?'
        DB      '番』です．',0DH,0AH,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
WORKST  DB      100H DUP(?)             ;GPAINT1 作業領域
ENDWRK  LABEL   BYTE
BUFFER  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS

        END     START
```

```
;###########################################################
;#                                                         #
;#      ＶＲＡＭ上の表示領域を設定する                          #
;#                                                         #
;###########################################################
GRAPH    GROUP    CODE,DATA,INTRPT,BUFFER
CODE     SEGMENT  PARA PUBLIC 'GRAPH'
         ASSUME   CS:CODE,DS:BUFFER,SS:STACK
START:
         MOV      AX,SEG BUFFER              ;作業領域セグメント
         MOV      DS,AX
;                                            ;GRAPH BIOS の初期設定
         MOV      WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
         MOV      WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
         MOV      AH,0
         INT      1DH
;                                            ;グラフ画面のモードを設定
         MOV      BYTE PTR DS:[100H],0              ;表示モード:カラー
         MOV      BYTE PTR DS:[101H],0              ;表示SW:表示あり,普通
         MOV      BYTE PTR DS:[102H],0              ;アクティブページ:0
         MOV      BYTE PTR DS:[103H],1              ;ディスプレイページ:1
         MOV      AH,1
         INT      1DH
;                                            ;描画領域を設定し,塗る
         MOV      WORD PTR DS:[100H],0              ;描画領域:
         MOV      WORD PTR DS:[102H],0              ;  (0,0)-(1024,749)
         MOV      WORD PTR DS;[104H],1024
         MOV      WORD PTR DS:[106H],749
         MOV      BYTE PTR DS:[108H],05H            ;描画領域色:05H
         MOV      BYTE PTR DS:[109H],0FFH           ;外枠は描かない
         MOV      AH,2
         INT      1DH
;                                            ;四辺形を描く
BOX:     MOV      WORD PTR DS:[100H],640            ;描画範囲:
         MOV      WORD PTR DS:[102H],500            ; (640,500)-(768,628)
         MOV      WORD PTR DS:[104H],768
         MOV      WORD PTR DS:[106H],628
         MOV      BYTE PTR DS:[108H],0AH            ;描画色:0FH
         MOV      BYTE PTR DS:[109H],02H            ;四辺形を描く/塗る
         MOV      BYTE PTR DS:[10AH],00H            ;ラインスタイルなし
         MOV      AH,7
         INT      1DH
;
         MOV      CX,0
                                             ;VRAM上の表示領域を設定する
SCROL:   MOV      DS:[100H],CX                      ;表示開始Y座標:0+n
         MOV      AH,10H
         INT      1DH
;
         INC      CX
         CMP      CX,186
         JL       SCROL
;                                                   ;メッセージを表示
         MOV      AX,SEG DATA
         MOV      DS,AX
         MOV      DX,OFFSET MSG
         MOV      AH,09H
         INT      21H
         MOV      AH,01H                            ;キー入力待ち
         INT      21H
;                                                   ;描画領域を設定し,塗る
         MOV      AX,SEG BUFFER                     ;作業領域セグメント
         MOV      DS,AX
         MOV      WORD PTR DS:[100H],0              ;描画領域:
         MOV      WORD PTR DS:[102H],0              ;  (0,0)-(1024,749)
         MOV      WORD PTR DS:[104H],1024
         MOV      WORD PTR DS:[106H],749
         MOV      BYTE PTR DS:[108H],0H             ;描画領域色:00H
         MOV      BYTE PTR DS:[109H],0FFH           ;外枠は描かない
         MOV      AH,2
         INT      1DH
EXIT:
         MOV      AH,4CH                            ;プログラムを終了
         INT      21H
CODE     ENDS

;#########   データ領域
DATA     SEGMENT  PARA PUBLIC 'GRAPH'
MSG      DB       '何かキーを押すと,グラフィック画面が消えます.'
         DB       0DH,0AH,'$'
GET      DB       3000 DUP(?)
DATA     ENDS

;#########   中断処理ルーチン
INTRPT   SEGMENT  PARA PUBLIC 'GRAPH'
         ASSUME   CS:INTRPT,DS:INTRPT
INTR     PROC     FAR
         RET
INTR     ENDP
INTRPT   ENDS

;#########   GRAPH BIOS 使用作業領域
BUFFER   SEGMENT  PARA PUBLIC 'GRAPH'
GBIOS    DB       256 DUP(?)                        ;GRAPH BIOS UCW
PARALST  DB       128 DUP(?)                        ;パラメータリスト
GWORK    DB       512 DUP(?)                        ;GRAPH BIOS 作業領域
BUFFER   ENDS

;#########   スタック領域
STACK    SEGMENT  PARA STACK 'STACK'
         DW       84 DUP (?)
STACK    ENDS
         END      START
```

```
;#####################################################################
;#                                                                  #
;#      グラフィック画面の表示を停止・開始する                       #
;#                                                                  #
;#####################################################################
GRAPH   GROUP   CODE,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER               ;作業領域セグメント
        MOV     DS,AX
;                                           ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                           ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[1ØØH],Ø          ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø          ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø          ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1          ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                           ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[1ØØH],Ø          ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø          ;   (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH         ;描画領域色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH       ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                           ;四辺形を描き,塗りつぶす
BOXFLL: MOV     WORD PTR DS:[1ØØH],3ØØ        ;描画範囲:
        MOV     WORD PTR DS:[1Ø2H],4ØØ        ;  (3ØØ,4ØØ)-(7ØØ,5ØØ)
        MOV     WORD PTR DS:[1Ø4H],7ØØ
        MOV     WORD PTR DS:[1Ø6H],5ØØ
        MOV     BYTE PTR DS:[1Ø8H],ØBH        ;描画色:ØBH
        MOV     BYTE PTR DS:[1Ø9H],Ø2H        ;四辺形/塗りつぶし
        MOV     BYTE PTR DS:[1ØAH],ØØH        ;スイッチ指定なし
        MOV     AH,7
        INT     1DH
;                                           ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG1
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                      ;キー入力待ち
        INT     21H
;                                           ;グラフ画面の表示を停止
        MOV     AH,12H
        INT     1DH
;                                           ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG2
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                      ;キー入力待ち
        INT     21H
;                                           ;グラフ画面の表示を開始
        MOV     AH,11H
        INT     1DH
;                                           ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø1H                      ;キー入力待ち
        INT     21H
;                                           ;描画領域を設定し,塗る
        MOV     AX,SEG BUFFER                 ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[1ØØH],Ø          ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø          ; (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØH         ;描画色:ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH       ;外枠は描かない
        MOV     AH,2
        INT     1DH
EXIT:
        MOV     AH,4CH                      ;プログラムを終了
        INT     21H
CODE    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      '何かキーを押すと,グラフィック画面が消えます.'
        DB      Ødh,Øah,'$'
MSG1    DB      '何かキーを押すと,グラフィック画面表示が停止'
        DB      'します.'
        DB      ØDH,ØAH,'$'
MSG2    DB      '何かキーを押すと,グラフィック画面表示が開始'
        DB      'されます.'
        DB      ØDH,ØAH,'$'
```

```
DATA    ENDS

;#########  GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)                ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                ;パラメータリスト
GWORK   DB      512 DUP(?)                ;GRAPH BIOS 作業領域
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS

        END     START
```

```
;#######################################################################
;#                                                                     #
;#      グラフィックBIOSの使用を終了する                               #
;#                                                                     #
;#######################################################################
GRAPH   GROUP   CODE,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER           ;作業領域セグメント
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;GRAPH BIOS の使用を終了
        MOV     AH,13H
        INT     1DH
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  GRAPH BIOS 使用作業領域
BUFFER  SEGMENT PARA PUBLIC 'GRAPH'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      84 DUP (?)
STACK   ENDS

        END     START
```

```
;*****************************************************************
;*                                                               *
;*      グラフィック画面の表示を停止・開始する                   *
;*                                                               *
;*****************************************************************
GRAPH   GROUP   CODE,INTRPT,BUFFER
CODE    SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:CODE,DS:BUFFER,SS:STACK
START:
        MOV     AX,SEG BUFFER            ;作業領域セグメント
        MOV     DS,AX
                                         ;GRAPH BIOS の初期設定
        MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     10H
                                         ;グラフ画面のモードを設定
        MOV     BYTE PTR DS:[100H],0         ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0         ;表示SW:表示する,数値
        MOV     BYTE PTR DS:[102H],0         ;アクティブページ  0
        MOV     BYTE PTR DS:[103H],1         ;ディスプレイページ1
        MOV     AH,1
        INT     10H
                                         ;描画領域を設定し,塗る
        MOV     WORD PTR DS:[100H],0         ;描画領域:
        MOV     WORD PTR DS:[102H],0         ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],8H        ;領域塗りつぶし色:08H
        MOV     BYTE PTR DS:[109H],0FFH      ;外枠は描かない
        MOV     AH,2
        INT     10H
                                         ;四辺形を描き,塗りつぶす
BOXFILL: MOV    WORD PTR DS:[100H],300       ;範囲指定
        MOV     WORD PTR DS:[102H],400       ; (300,400)-(700,500)
        MOV     WORD PTR DS:[104H],700
        MOV     WORD PTR DS:[106H],500
        MOV     BYTE PTR DS:[108H],08H       ;描画色:08H
        MOV     BYTE PTR DS:[109H],02H       ;四辺形/塗りつぶし
        MOV     BYTE PTR DS:[10AH],00H       ;スイッチ設定なし
        MOV     AH,?
        INT     10H
                                         ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG1
        MOV     AH,09H
        INT     21H
                                         ;キー入力待ち
        MOV     AH,?
        INT     21H
                                         ;グラフ画面の表示を停止
        MOV     AH,12H
        INT     10H
                                         ;メッセージを表示
        MOV     AX,SEG DATA
        MOV     DS,AX
        MOV     DX,OFFSET MSG2
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                   ;キー入力待ち
        INT     21H
                                         ;グラフ画面の表示を開始
        MOV     AH,11H
        INT     10H
                                         ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
                                         ;作業領域セグメント
        MOV     DS,AX
        MOV     WORD PTR DS:[100H],0         ;描画領域:
        MOV     WORD PTR DS:[102H],0         ; (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
                                             ;描画色:08H
                                             ;外枠は描かない
        INT     10H
                                         ;プログラムを終了
        INT     21H
CODE    ENDS
INTRPT  SEGMENT PARA PUBLIC 'GRAPH'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
INTR    ENDP
INTRPT  ENDS
DATA    SEGMENT PARA PUBLIC 'GRAPH'
MSG     DB      'グラフィック画面表示が停止'
        DB      0DH,0AH,'$'
MSG2    DB      'グラフィック画面表示が開始'
        DB      'されます'
```

# 第5章

# ディスクBIOS

## 5.1　DISK BIOS 共通情報

### 5.1.1　DISK BIOS コマンドの一般形式

この項はディスク全般の共通事項であり 1MBFD, 640KBFD, 固定ディスクの各 BIOS コマンドの項とあわせて読むこと．

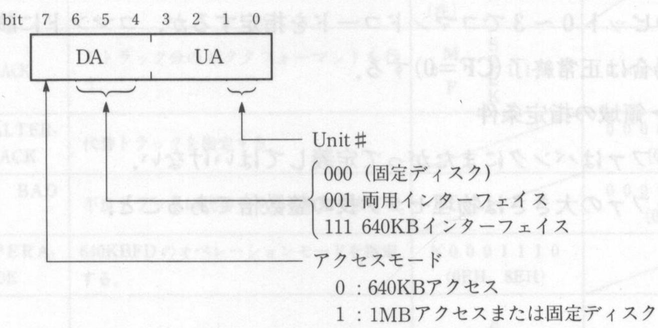
**(1) 内部割り込みコード**

1MBFD, 640KBFD, 固定ディスクの場合：INT 1BH

**(2) 入力**

・AH ← BIOS コマンド識別コード(BIOS コマンド一覧参照)

・AL ←デバイスタイプ識別コード(DA：Device　Address)／ユニット番号(UA：Unit Address)




注：PC-98XL/XA ハイレゾリューションモード時の, DA/UA とデバイスの対応は次のとおり．

90H～93H…… 両用インターフェイスによる 1MB ディスクアクセス

80H～81H…… 固定ディスクアクセス

70H～73H…… 640KB インターフェイスによる 640KB ディスクアクセス

10H～13H…… 両用インターフェイスによる 640KB ディスクアクセス

・BX ←データ長(DTL)
・CH ←セクタ長(N)
・CL ←シリンダ番号(C)
・DH ←ヘッド番号(H)
・DL ←セクタ番号(R)

（CH・CL・DH・DL の4項目）これらをあわせてID情報と呼ぶ(略称IDR)

・ES ←データバッファ領域先頭アドレス(セグメントアドレス)
・BP ←データバッファ領域先頭アドレス(オフセットアドレス)

**(3) 出力**

・CF ←終了条件
　0：正常終了
　1：異常終了
・AH ←ステータス情報(ステータス情報一覧参照)
・出力情報として使用されるレジスタ，およびフラグ以外はすべて保証される.
・システム共通域中DISK_RESULT(564H～583H)の8バイトエントリ，またはF2DD_RESULT(5D0H～5DFH)の16バイトエントリにFDCからのリザルトステータス情報を格納.

**(4) BIOSコマンド使用上の注意**

① コマンドコードでないコードの扱い

AHレジスタのビット0～3でコマンドコードを指定するが，コマンドに該当しないコードが指定された場合は正常終了(CF=0)する.

② データバッファ領域の指定条件

a）データバッファはバンクにまたがって定義してはいけない.
b）データバッファの大きさは物理セクタ長の整数倍であること.

## 5.1.2 BIOS コマンド一覧

| コマンド名 | 機　　能 | コマンドコード AH 1MBFD / 640KBFD | コマンドコード AH 固定ディスク |
|---|---|---|---|
| READ DATA | ディスク上のデータを読み取る. | MT MF $\bar{r}$ SEEK 0 1 1 0 | × × $\bar{r}$ × 0 1 1 0 |
| WRITE DATA | ディスク上にデータを書き込む. | MT MF $\bar{r}$ SEEK 0 1 0 1 | × × $\bar{r}$ × 0 1 0 1 |
| SEEK | 指定された物理番号のシリンダへアームを移動し，ヘッドを選択する. | × × $\bar{r}$ 1 × × × ×<br>(10H) | |
| RECALIBRATE | 物理番号のシリンダ0へアームを移動する. | 0 0 $\bar{r}$ 0 0 1 1 1<br>(07H) | × × $\bar{r}$ × 0 1 1 1 |
| FORMAT DRIVE | ディスクのフォーマットを行う. | | 1 × $\bar{r}$ × 1 1 0 1 |
| INITIALIZE | コントローラのイニシャライズを行う. | 0 0 0 0 0 0 1 1<br>(03H) | 0 0 0 0 0 0 1 1<br>(03H) |
| VERIFY | ディスク上のデータを読み取るが，主記憶には転送しない. | MT MF $\bar{r}$ SEEK 0 0 0 1 | × × $\bar{r}$ × 0 0 0 1 |
| SENSE | デバイスの状態，または属性を読み取る. | n 0 $\bar{r}$ 0 0 1 0 0<br>(04H) | 0 0 0 0 0 1 0 0<br>(04H) |
| READ ID | トラック上のエラーのない ID を読み取る. | 0 MF $\bar{r}$ SEEK 1 0 1 0 | |
| WRITE DELETED DATA | DDAM (Deleted Data Address Mark)付データを書き込む. | MT MF $\bar{r}$ SEEK 1 0 0 1<br>(注) | |
| FORMAT TRACK | 1トラック分のセクタフォーマットを行う. | 0 MF $\bar{r}$ SEEK 1 1 0 1 | 0 × $\bar{r}$ × 1 1 0 1 |
| ASSIGN ALTERNATE TRACK | 代替トラックを指定する. | | 0 0 0 0 1 0 0 0<br>(08H) |
| FORMAT BAD TRACK | 不良トラックに代替トラックを割り当てる. | | 0 0 0 0 1 0 1 1<br>(0BH) |
| SET OPERATION MODE | 640KBFD のオペレーションモードを指定する. | 1/0 0 0 0 1 1 1 0<br>(0EH, 8EH) | |
| READ DELETED DATA | DDAM 付データを読み取る. | MT MF $\bar{r}$ SEEK 1 1 0 0<br>(注) | |
| READ DIAGNOSTIC | ID/DATA 部のエラーが検出されても読み取りを続行する. | 0 MF $\bar{r}$ SEEK 0 0 1 0<br>(注) | |
| RETRACT | ヘッドを不使用シリンダへ移動する (CSS ゾーン) | | × × $\bar{r}$ × 1 1 1 1 |

注：1MBFD のみ

MT　1：マルチトラック指定
　　0：シングルトラック

MF　1：倍密度指定
　　0：単密度

SEEK　1：SEEK 動作を伴う
　　　0：SEEK 動作なし

$\bar{r}$　1：リトライ指定
　　0：リトライなし

×：0または1
　　どちらでもよい

## 5.1.3　ステータス情報一覧

| CF | AH | | 説明 | |
|---|---|---|---|---|
| | 16進表示 | ビット | 略称 | 内容 |
| 0 | 00H | 0 0 0 0 ×××× | NT | Normal end |
| 0 | 00H | | RY | Ready (Senseコマンド) |
| 0 | 10H | 0 0 0 1 ×××× | CM | Control Mark (1MBFD) |
| 0 | 10H | | WP | Write Protect (Senseコマンド) |
| 1 | 20H | 0 0 1 0 ×××× | DB | DMA Boundary (XAにはない) |
| 1 | 30H | 0 0 1 1 ×××× | EN | End of cylinder |
| 1 | 40H | 0 1 0 0 ×××× | EC | Equipment Check |
| 1 | 50H | 0 1 0 1 ×××× | OR | OverRun |
| 1 | 60H | 0 1 1 0 ×××× | NR | Not Ready |
| 1 | 70H | 0 1 1 1 ×××× | NW | Not Writable |
| 1 | 80H | 1 0 0 0 ×××× | SE | Seek Error |
| 1 | 90H | 1 0 0 1 ×××× | TO | TimeOut |
| 1 | A0H | 1 0 1 0 ×××× | DE | DataError (ID) |
| 1 | B0H | 1 0 1 1 ×××× | DD | DataError (Data) |
| 1 | C0H | 1 1 0 0 ×××× | ND | No Data |
| 1 | D0H | 1 1 0 1 ×××× | BC | Bad Cylinder |
| 1 | E0H | 1 1 1 0 ×××× | MA | Missing Address mark (ID) |
| 1 | F0H | 1 1 1 1 ×××× | MD | Missing Address mark (Data) |
| 0 | 01H | ×××× 0 0 0 1 | | 両面媒体がセットされている |

注：ステータスのビット7～5は，CF=0のとき000，CF=1のとき000以外となる．
　　ビット3～0はセンスコマンドで装置種別が通知される(CF=0のとき有効)．

## 5.1.4 システム共通域一覧

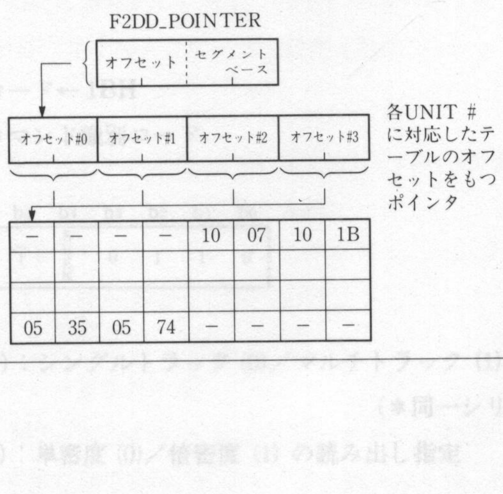
| システム共通域<br>シンボル名・絶対番地 | 用　途 | 備　考 |
|---|---|---|
| DISK_RESET<br>(492H) | ・各ユニットのbitがONの時，次のアクセスでリキャリブレイトを実行する．リキャリブレイト実行後OFFにする．<br>b7 b0<br>W W W W X X X X<br>640KBFD 1MB FD<br>UNIT # → 3 2 1 0 3 2 1 0 ← UNIT #<br>・リキャリブレイト実行条件<br>①Time Outが発生し，FDCがリセットされた時．<br>②両用タイプドライブでアクセスモードを変更した時． | BIOSのStatus情報にTime Outが通知された時，および上位ソフトウェアがアクセスモードを変更した時リセットが行われる．<br>リセットされるとカレントシリンダアドレスが失われるため，シークbit offでのアクセスは注意が必要である． |
| F2HD_MODE<br>(493H) | ・両用タイプFDのオペレーションモードの情報を格納する．<br>b7 b0<br>UNIT #0 / UNIT #1 / UNIT #2 / UNIT #3 ― 0：片面モード　1：両面モード<br>UNIT #0 / UNIT #1 / UNIT #2 / UNIT #3 ― 0：48tpiモード　1：96tpiモード | 初期値は，(0FFH)である． |
| DISK_EQUIP2<br>(494H) | ・両用タイプインタフェースが640KBモードとなっているときに，接続されているFDDの状態を示す．<br>1：接続されている<br>0：接続されていない<br>b7 b0<br>UNIT #0<br>UNIT #1<br>UNIT #2<br>UNIT #3 | |
| DISK_EQUIP<br>(55CH・55DH) | ・各装置タイプ毎のINITIALIZEを行うと，接続ユニットの状態がセットされる．<br>1：接続されている<br>0：接続されていない<br>b7 b0 b15 b8<br>0 0 0 0 X X X X W W W W 0 0 Z Z<br>1 0 固定ディスク UNIT #<br>1MBFD→ 3 2 1 0 3 2 1 0 ←640KBFD<br>UNIT# UNIT # | 1MBとしてセットされるFD装置は両用タイプの場合があり，Senseコマンドで確認が必要． |

| システム共通域<br>シンボル名・絶対番地 | 用　　途 | 備　　考 |
|---|---|---|
| DISK_INT<br>(55EH・55FH) | ・割り込みのあったユニットに対応するビットをオン(1)にし，上位にその結果を通知したときオフ(0)にする．<br>b7 ... b0 b15 ... b8<br>0 0 0 0 X X X X W W W W 0 0 0 Z<br>1MBFD UNIT#→ 3 2 1 0 (X)　固定ディスク (Z)<br>640KBFD UNIT#→ 3 2 1 0 (W) | |
| DISK_RESULT<br>(564H～583H) | ・FDCから通知されるリザルトステータス情報を格納する．<br>・各デバイス毎に8バイトエントリを持っている．4デバイス(ユニット)分ある．<br>+24 #3<br>+16 #2<br>+8 #1<br>#0 (b7 b6 b5 b4 b3 b2 b1 b0)<br>+0: IC1 IC0 SE EC NR HD US1 US0　ST0<br>+1: EN 0 DE OR — ND NW MA　ST1(注)<br>+2: 0 CM DD NC SH SN BC MD　ST2<br>+3: 0～76　C(シリンダ#)<br>+4: 0～1　H(ヘッド#)<br>+5: 1～26　R(セクタ#)<br>+6: 00H～03H　N(セクタ内のデータ長)<br>+7: 0～76　NCN(現在のシリンダ#) | 注：ST0のSE＝0のときのみST1として，SE＝1のときPCN(プリゼントシリンダナンバ)として使用． |
| DISK_BOOT<br>(584H) | ブート装置のデバイス種別・ユニット番号(DA／UA)が格納される．この領域内はBIOSがワークとして参照するものである． | |
| DISK_STATUS<br>(585H) | 固定DISKコントローラから送られるステータスが格納される．<br>b7 b6 b5 b4 b3 b2 b1 b0<br>[b7～b5: LVN] X X X [b1: パリティエラー] [b0: エラー] | |
| DISK_SENSE<br>(586H～589H) | ・固定ディスクのRequest Senseコマンドで通知されるSense Status Bytesを格納する．<br>b7 ... b0<br>0: センスバイト<br>1: 0 0 ｜ 論理アドレス(H)<br>2: 論理アドレス(M)<br>3: 論理アドレス(L)<br>(1～3: 論理アドレス) | |

| システム共通域<br>シンボル名・絶対番地 | 用　　　途 | 備　　考 |
|---|---|---|
| F2DD_MODE<br>(5CAH) | ・オペレーションモード指定<br>・このエリアはシステム初期化後FFHとなる．<br>b7 b6 b5 b4 b3 b2 b1 b0<br>W W W W W W W W<br>UNIT# → 3 2 1 0 3 2 1 0 ← UNIT#<br>0：48tpi　0：片面モード<br>1：96tpi　1：両面モード | 640KBFD用 |
| F2DD_COUNT<br>(5CBH) | ・640KBFDのモータをoffにするための初期カウンタ値．単位は約100ms<br>・システム初期化，または640KBFDの初期化コマンド実行後，このエリアの値は150₁₀(約15秒)に設定される．<br>b7 b6 b5 b4 b3 b2 b1 b0<br>01H～0FFH | |
| F2DD_POINTER<br>(5CCH～5CFH) | ・コマンドのNバイト(セクタ長：CHレジスタで指定する)に対応するIDのNバイト，EOT，Read／Write時のGPLおよびFormat時のGPLを与えるTableへのポインタ．<br>・システム初期化後，このポインタは640KBFDBIOSルーチン内のテーブルを指している．<br>BIOSが持つテーブルの値は次のとおりである．（下表） | |

| MFM | | | | FM | | | | |
|---|---|---|---|---|---|---|---|---|
| R／W | | Format | | R／W | | Format | | |
| EOT | GPL | EOT | GPL | EOT | GPL | EOT | GPL | |
| − | − | − | − | 10 | 07 | 10 | 1B | ←N＝0 |
| 10 | 0E | 10 | 33＊1 | 09 | 0E | 09 | 2A | ←N＝1 |
| 09 | 2A | 09 | 50＊2 | 05 | 1B | 05 | 3A | ←N＝2 |
| 05 | 35 | 05 | 74 | − | − | − | − | ←N＝3 |

−：未定義
＊1：DISK-BASIC (9800シリーズ)
＊2：MS-DOS
その他：IBM 8″FD準拠



| システム共通域<br>シンボル名・絶対番地 | 用　　　　途 | 備　　　考 |
|---|---|---|
| F2DD_RESULT<br>(5D0H<br>〜<br>5DFH) | ・640KBFD用リザルトステータス格納域<br><br>005D0H　$ST_0$<br>$ST_1$<br>$ST_2$<br>C<br>H<br>R<br>N<br>モータオフカウンタ<br>005D8H　ST0<br>$PCN_0$<br>ST0<br>$PCN_1$<br>ST0<br>$PCN_2$<br>ST0<br>$PCN_3$<br><br>($ST_0$〜N) Read／Write系コマンドのリザルトステータスがセットされる<br><br>(モータオフカウンタ) I／O終了後，F2DD_COUNTの値をここへコピーし，タイマネーブルにし，タイマINT毎にデクリメントする．0になったときモータオフし，タイマをディスエーブルする<br><br>(005D8H〜) シーク／リキャリブレイトコマンドのリザルトステータスがセットされる | ST0について，次の操作を行う．<br>(1) MOTOR OFF時の処理<br>MOTOR OFF時にシステム共通域の各ユニット毎の情報(STQ)のNRビット(Not Ready…$D_3$)をすべてONにする．<br>(2) MOTOR ON時の処理<br>MOTOR OFFタイマカウンタをチェックする前に，上記のNRビットをチェックしONの時，Waitループを実行する(実行後NRビットはOFFにする)． |

## 5.2 1MB フロッピーディスク

PC-98XA/XL には標準で 1MB/640KB 両用タイプのインターフェイスが内蔵されており，DA/UA＝90H～93H としてコマンドを発行すると，1MB フロッピーディスクへのアクセスとなる．DA/UA＝10H～13H の場合，両用インターフェイスは 640KB ディスクへのアクセスとみなす．

両用タイプのインターフェイスにおいては，装置の識別は SENSE コマンドによって行い，ステータスの b3 ビットが ON のとき，両用タイプであることを示す(SENSE コマンドは，DA/UA が 9×H でも 1×H でも有効)．

DA/UA が 10H から 13H の場合，動作(コマンドおよびパラメータ)は 640KB インターフェイスとまったく同様になる．

### 5.2.1 データの読み出し(READ DATA)

(1) 機能

指定されたデバイス種別・ユニット番号(DA/UA＝90H～93H)を持つデバイスの，指定された実行開始セクタ(ID 情報；シリンダ番号(C)，ヘッド番号(H)，セクタ番号(R)，セクタ長(N))から，指定されたメモリ領域(データバッファ領域の先頭アドレス)へ，指定された長さ(DTL)のデータを読み出す．

現在選択されているシリンダ位置のトラックから読み出す場合と，シーク動作を行い，トラックの選択を行ってから読み出す場合との選択が可能である．

このコマンドはシングルトラック／マルチトラックの読み出し選択指定(MT-Multi Track)および，単密度(FM モード)／倍密度(MFM モード)の読み出し選択指定(MF-Multiple Frequency modulation)が可能である．また，エラー発生時に，8 回のリトライを行う(r ビット＝0 の場合)．リトライをしない指定($\overline{r}$ ビット＝1)も可能である．

(2) 入力

・内部割り込みコード←1BH

・AH←BIOS コマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| MT | MF | $\overline{r}$ | SEEK | 0 | 1 | 1 | 0 |

$b_7$(MT)：シングルトラック (0)／マルチトラック (1) *の読み出し指定

(*同一シリンダの両面トラックのみ)

$b_6$(MF)：単密度 (0)／倍密度 (1) の読み出し指定

$b_5$($\overline{r}$)：8回リトライする (0)／リトライなし (1) の指定

$b_4$(SEEK)：シーク動作が必要 (1)／現在のトラック位置 (0) からの読み出し指定

・AL ←デバイス種別・ユニット番号(DA/UA)(90H～93H)

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | $x_2$ | $x_1$ |

($b_1$, $b_0$) 接続ディスク装置のドライブに対応
(00←0～11←3)

・BX ←データ長(DTL)(バイト単位)

・CL ←シリンダ番号(C)(0～76)

・DH ←ヘッド番号(H)(0～1)

・DL ←セクタ番号(R)(1～26, 15, 8)

・CH ←セクタ長(N)(00H～03H)

(CL・DH・DL・CH) これらをあわせてID情報と呼ぶ(略称IDR)

| CH | 00 | 01 | 02 | 03 |
|---|---|---|---|---|
| セクタ長(バイト／セクタ) | 128 | 256 | 512 | 1024 |

・ES/BP ←データバッファ領域の先頭アドレス

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　容 |
|---|---|---|---|
| 0001××××<br>(10H) | CM | Control Mark | DDAM (Deleted Data Address Mark)を検出した．そのセクタを読み出した後，正常終了する． |
| 0010××××<br>(20H) | DB | DMA Boundary | メモリアドレスが奇数番地から始まるように指定した場合の異常終了ステータス(使われていない)． |
| 0011××××<br>(30H) | EN | End of Cylinder | 1回の動作の転送容量を超えてデータ長(DTL)を指定した場合の異常終了ステータス． |
| 0100××××<br>(40H) | EC | Equipment Check | デバイスが異常か，リキャリブレイトコマンド実行時に，一定時間内にトラック0信号を確認できなかった場合の異常終了ステータス． |
| 0101××××<br>(50H) | OR | Over Run | セクタのデータをメモリに転送しているとき，一定時間内にデータ転送が終了できなかった場合の異常終了ステータス． |
| 0110××××<br>(60H) | NR | Not Ready | ユニットがノットレディ状態の場合の異常終了ステータス． |
| 1010××××<br>(A0H) | DE | Data Error | ID読み出し時にCRCエラーが発生した場合の異常終了ステータス． |
| 1100××××<br>(C0H) | ND | No Data | トラック内に，指定されたセクタが見つからなかった場合の異常終了ステータス． |
| 1110××××<br>(E0H) | MA | Missing Address mark | トラック内に，指定されたセクタが見つからず，かつIDが1個もなかった場合の異常終了ステータス．(注) |

注：指定されたセクタのIDを検出した後，データ部のアドレスマークを一定時間内に検出できなかった場合にもMAで異常終了する．

・システム共通域(DISK_RESULT 564H〜583Hの8バイトエントリ)

←FDCからのリザルトステータス情報を格納

**(4) 処理**

① DA/UAに対応するシステム共通域DISK_RESULTの内容を調べる．

② SEEKビットが1(オン)かどうかを判定する．

オンならばシリンダ番号Cによってシーク動作を行う．シーク動作が終了したか，またはSEEKビットが0(オフ)ならば③の処理を行う．

③ DMAを起動する．

④ FDCにREADコマンドを発行する．

(マルチトラックMT，MFMモードはBIOSコマンドに従う．DDAM(Deleted Data Address Mark)を検出した場合は，そのセクタの読み取り後，コマンドの実行を正常終了する．)

⑤ FDCに対して，READコマンドのパラメータを順次送出する．

C(シリンダ#)，H(ヘッド#)，R(セクタ#)，N(セクタ長)

⑥ 割り込み信号の入力を待つ．

入力処理の終了を待ち，リザルトステータス情報を読み出し(ST0，ST1，ST2)，実行終了セクタのID情報(C，H，R，N)の読み出しを行い，システム共通域のRESULT_DATA，BIOSコマンド出力条件の設定を行う．

⑦ DMAチャネルを閉じる．

## (5) その他

a）1回の動作の転送容量はMT(マルチトラック)ビット，MF(倍密度指定)ビット，N(セクタ長)に関係する．

| MT | MF | N | 転　送　容　量(バイト) | | 最終セクタ |
|---|---|---|---|---|---|
| 0 | 0 | 00H | 1～128×n | n＝1～26 | ヘッド0のセクタ26または<br>ヘッド1のセクタ26 |
| | 1 | 01H | 1～256×n | | |
| 1 | 0 | 00H | 1～128×n | n＝1～52 | ヘッド1のセクタ26 |
| | 1 | 01H | 1～256×n | | |
| 0 | 0 | 01H | 1～256×n | n＝1～15 | ヘッド0のセクタ15または<br>ヘッド1のセクタ15 |
| | 1 | 02H | 1～512×n | | |
| 1 | 0 | 01H | 1～256×n | n＝1～30 | ヘッド1のセクタ15 |
| | 1 | 02H | 1～512×n | | |
| 0 | 0 | 02H | 1～512×n | n＝1～8 | ヘッド0のセクタ8または<br>ヘッド1のセクタ8 |
| | | 03H | 1～1024×n | | |
| 1 | 0 | 02H | 1～512×n | n＝1～16 | ヘッド1のセクタ8 |
| | 1 | 03H | 1～1024×n | | |

注：nの値はプログラマブルで，EOTの値で定義される．

b）BIOS コマンド実行を正常に終了したときに，システム共通域の RESULT_DATA に設定される IDR(C，H，R，N)の状態は次表のようになる．

| MT | EOT | 最終バイトの転送に関したセクタ | IDR の 状 態 C | H | R | N |
|---|---|---|---|---|---|---|
| 0 | 1AH<br>0FH<br>08H | ヘッド0のセクタ 1～25<br>〃 1～14<br>〃 1～7 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 1AH<br>0FH<br>08H | ヘッド0のセクタ 26<br>〃 15<br>〃 8 | C＋1 | もとのまま | R＝01(注) | もとのまま |
| | 1AH<br>0FH<br>08H | ヘッド1のセクタ 1～25<br>〃 1～14<br>〃 1～7 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 1AH<br>0FH<br>08H | ヘッド1のセクタ 26<br>〃 15<br>〃 8 | C＋1 | もとのまま | R＝01(注) | もとのまま |
| 1 | 1AH<br>0FH<br>08H | ヘッド0のセクタ 1～25<br>〃 1～14<br>〃 1～7 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 1AH<br>0FH<br>08H | ヘッド0のセクタ 26<br>〃 15<br>〃 8 | もとのまま | 下1ビットを反転する | R＝01(注) | もとのまま |
| | 1AH<br>0FH<br>08H | ヘッド1のセクタ 1～25<br>〃 1～14<br>〃 1～7 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 1AH<br>0FH<br>08H | ヘッド1のセクタ 26<br>〃 15<br>〃 8 | C＋1 | 下1ビットを反転する | R＝01(注) | もとのまま |

注：ヘッドの選択状態(ST0 の HD ビットも含む)は最終バイトの転送に関係したセクタに属するトラックのままである．

c）動作終了後，ヘッドをすぐにはアンロード状態にせず，一定時間経過後(約1回転＝160msec)アンロード状態にする．したがってこの時間内に以前と同じユニットに対してリードまたはライト系のコマンドが実行された場合，ヘッドロードの安定時間待ち(約50msec)を節約することが出来る．

ただし，この時間以内に他のデバイスに対しての動作が行われた場合には，以前のデバイスのヘッドはアンロード状態になる．

また，動作が起動された方のユニットでは自動的にロード安定時間待ちが行われる．

## 5.2.2 データの書き込み(WRITE DATA)

### (1) 機能

指定されたデバイス種別・ユニット番号(DA/UA＝90H～93H)を持つデバイスの，指定された実行開始セクタ(ID情報；シリンダ番号(C)，ヘッド番号(H)，セクタ番号(R)，セクタ長(N))から，指定されたメモリ領域(データバッファ領域の先頭アドレス)の，指定された長さ(DTL)のデータを書き込む．

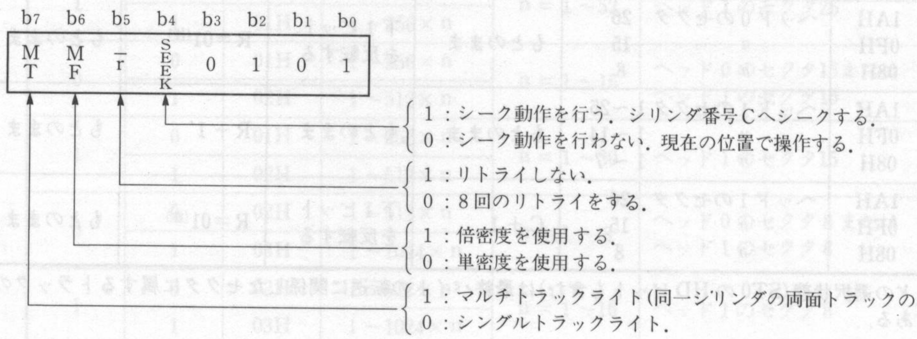
現在選択されているシリンダ位置のトラックへ書き込む場合と，シーク動作を行い，トラックの選択を行ってから書き込む場合との選択が可能である．また，単密度(FMモード)／倍密度(MFMモード)の書き込み選択指定(MF-Multiple Frequency modulation)も可能である．

注：マルチトラック指定は使用しないこと．

### (2) 入力

・内部割り込みコード←1BH

・AH ← BIOSコマンド識別コード




注：MTビットは0で使用すること

・AL ←デバイス種別・ユニット番号(DA/UA)(90H～93H)
・BX ←データ長(DTL)(バイト単位)
・CH ←セクタ長(N)(00H～03H)(「5.2.1」参照)
・CL ←シリンダ番号(C)(0～76)
・DH ←ヘッド番号(H)(0～1)
・DL ←セクタ番号(R)(1～26, 15, 8)
・ES/BP ←データバッファ領域の先頭アドレス

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　　容 |
|---|---|---|---|
| 30H | EN | End of Cylinder | 「5．2．1」参照 |
| 40H | EC | Equipment Check | データの書き込み時，デバイスのFAULT状態を検出した場合の異常終了ステータス． |
| 50H | OR | Over Run | セクタのデータ部にメモリからデータ転送をしているとき，一定時間内にデータを転送できなかった場合の異常終了ステータス．そのセクタを書き込み後，ORとなる． |
| 60H | NR | Not Ready | 「5．2．1」参照 |
| 70H | NW | Not Writable | コマンド実行開始時，WRITE PROTECTED信号がオンの場合の異常終了ステータス． |
| A0H | DE | Data Error | 1セクタのIDを読み取るごとにCRCバイトをチェックし，いずれかのセクタでCRCエラーが生じた場合の異常終了ステータス． |
| C0H | ND | No Data | 「5．2．1」参照 |
| E0H | MA | Missing Address mark | 「5．2．1」参照 |

注：MTビットを指定してトラックにまたがるI/Oを実行するとデータをこわすことがある．

・システム共通域(DISK_RESULT 564H～583Hの8バイトエントリ)

←FDCからのリザルトステータス情報を格納

**(4) 処理**

「5．2．1」と同様の処理を行う．

注：データ長(DTL)で示すバイト数のデータの書き込みがセクタの途中で終了した場合は，そのセクタの残りのバイトには00Hを書き込んで正常終了する．

## 5．2．3　シークを行う(SEEK)

**(1) 機能**

指定されたデバイス種別・ユニット番号(DA/UA)に対して，指定されたシリンダ物理番号(C)までシークさせる．

なお，読み出し(READ)／書き込み(WRITE)用のBIOSコマンドには，シーク動作を同時に行うことが可能になっている．それぞれのBIOSコマンド識別コードの $b_4$ ビット(10H)をオン(1)にするとシーク動作を伴い，オフ(0)にすると現在のシリンダを対象とした動作を行う．

(2) 入力

・内部割り込みコード←1BH

・AH ←10H(BIOSコマンド識別コード)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | × | $\bar{r}$ | 1 | × | × | × | × |

・AL ←シークを行うデバイス種別・ユニット番号(DA/UA)(90H～93H)

・CL ←シークを行うシリンダ番号(C)(0～76)

(3) 出力

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

40H：Equipment Check (EC)

デバイスからFault信号を受取ったとき，ECによる異常終了となる.

60H：Not Ready (NR)

FDCがシークコマンドを開始する時，またはデバイスがシーク動作実行中，デバイスがノットレディ状態であると，NRによる異常終了となる.

・システム共通域DISK_RESULT(564H～586H)の8バイトエントリ

←現在のシークアドレス(シリンダ番号)をPCNとして格納.

(4) 処理

① FDCにシークコマンド(10H)を送出する.

② FDCにシークコマンドに対するパラメータを送出する.

シークを行うユニット番号，ヘッド番号を渡す.

③ シーク動作の終了を待つ. シーク実行中はBUSYでないため, ユニット4台まで同時動作が可能.

④ シーク動作が終了すると出力のステータス情報を読み出す.

## 5.2.4 シリンダ0へシークする(RECALIBRATE—リキャリブレイト)

### (1) 機能

シリンダ物理番号0(デバイスから，トラック0信号を検出するまで)へシークさせる．

シーク動作は，シリンダ物理番号0の方向へ1シリンダずつ行い，デバイスからのトラック0信号を検出するまでくり返す．

### (2) 入力

・内部割り込みコード←1BH

・AH ←07H(BIOSコマンド識別コード)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $\overline{r}$ | 0 | 0 | 1 | 1 | 1 |

・AL ←デバイス種別・ユニット番号(DA/UA)(90H～93H)

### (3) 出力

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

40H：Equipment Check (EC)

デバイスからのトラック0信号がオン (1) になるまで77回くり返すが，検出できなかった場合，ECによる異常終了となる．

60H：Not Ready (NR)

「5.2.3」参照

## 5.2.5 トラックのフォーマット(FORMAT TRACK)

### (1) 機能

指定されたセクタ長(N)，トラック当りのセクタ数(SC)，ギャップ長(GPL)，データ部に書き込むデータパターン(D)などに従って1トラック分フォーマット書き込みを行う．

セクタのID部にデータを書き込むときには，指定されたデータバッファ領域の内容(C，H，R，Nの4バイト×セクタ数)を使用する．したがって，セクタシーケンスIDや不良シリンダIDなども書き込むことが可能である(C，H，R，Nの指定による)．

各セクタのデータ部には，1バイトのデータDをNで指定される長さ分だけくり返し書き込む．

**(2) 入力**

・内部割り込みコード←1BH

・AH ← BIOS コマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | MF | $\overline{r}$ | SEEK | 1 | 1 | 0 | 1 |

- SEEK（b4）
  - 1：シーク動作を行う．シリンダ番号Cへシークする．
  - 0：シーク動作を行わない．現在のシリンダ位置で操作する．
- $\overline{r}$（b5）
  - 1：リトライしない．
  - 0：8回のリトライをする．
- MF（b6）
  - 1：倍密度を使用する．
  - 0：単密度を使用する．

・AL ←デバイス種別・ユニット番号(DA/UA)(90H～93H)

・BX ←データ長(DTL)(バイト単位)

・CH ←セクタ長(N)(00, 01, 02, 03)

・CL ←シリンダ番号(C)(0～76)(AH の SEEK ビットが1のとき有効)

・DH ←ヘッド番号(H)(0～1)

・DL ←データ部への書き込みデータパターン(D)

・ES/BP ←データバッファ領域の先頭アドレス

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　容 |
|---|---|---|---|
| 40H | EC | Equipment Check | 書き込み終了時に Fault 状態を検出した． |
| 50H | OR | Over Run | セクタの ID 部に書き込み時，一定時間内にデータ転送ができなかった． |
| 60H | NR | Not Ready | デバイスがノットレディ状態になった． |
| 70H | NW | Not Writable | 「5．2．2」参照 |

### (4) 処理

①DA/UAで指定されるユニットの，現在選択されているシリンダのHバイトで示されるヘッドの面にあるトラックにフォーマットを書き込む．

②セクタのID部の書き込みは，指定されているデータバッファ領域の先頭アドレス(ES : BP)から，BXのデータ長までのデータバッファ上に，セクタのID部ごとのC，H，R，Nの4バイトエントリをトラックのセクタ数分だけ展開しておき，これを書き込む((5) を参照)．

③セクタのデータ部の書き込みは，指定されたDL(データパターンD)の内容を，指定されたCH(セクタ長N)の長さ分だけセクタごとに繰り返す．

### (5) セクタのID部をフォーマットするための準備

①パラメータの指定方法

| 指定方法 | | 指定内容 | | トラック当りのセクタ数 | ギャップ長 | IDフォーマット用データバッファ長 | 備考 |
|---|---|---|---|---|---|---|---|
| MFビット | Nセクタ長 | 密度 | セクタ当りのバイト数 | SC | GPL | (DTL) | |
| 0 | 00 | 単密度 | 128 | 1AH (26) | 1BH | 104 | PC-9800シリーズ $N_{88}$-BASIC (シリンダ0，HEAD0) MS-DOS(8″1s) |
| 0 | 01 | FMモード | 256 | 0FH (15) | 2AH | 60 | |
| 0 | 02 | | 512 | 08H (8) | 3AH | 32 | |
| 1 | 01 | 倍密度 | 256 | 1AH (26) | 36H | 104 | PC-9800シリーズ $N_{88}$-BASIC (シリンダ0，HEAD0以外) |
| 1 | 02 | MFMモード | 512 | 0FH (15) | 54H | 60 | |
| 1 | 03 | | 1024 | 08H (8) | 74H | 32 | MS-DOS |

注：DTLは上表の値以上であればよい．256をとるようにする．

② セクタシーケンスに従った論理セクタ番号

a）26セクタ／トラックの場合

| セクタシーケンス | 物理セクタ番号(16進) | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | |
| | 論理セクタ番号(16進) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | ← CP/M, MS-DOS |
| 02 | 01 | 03 | 05 | 07 | 09 | 0B | 0D | 0F | 11 | 13 | 15 | 17 | 19 | 02 | 04 | 06 | 08 | 0A | 0C | 0E | 10 | 12 | 14 | 16 | 18 | 1A | |
| 03 | 01 | 04 | 07 | 0A | 0D | 10 | 13 | 16 | 19 | 02 | 05 | 08 | 0B | 0E | 11 | 14 | 17 | 1A | 03 | 06 | 09 | 0C | 0F | 12 | 15 | 18 | |
| 04 | 01 | 05 | 09 | 0D | 11 | 15 | 19 | 02 | 06 | 0A | 0E | 12 | 16 | 1A | 03 | 07 | 0B | 0F | 13 | 17 | 04 | 08 | 0C | 10 | 14 | 18 | |
| 05 | 01 | 06 | 0B | 10 | 15 | 1A | 02 | 07 | 0C | 11 | 16 | 03 | 08 | 0D | 12 | 17 | 04 | 09 | 0E | 13 | 18 | 05 | 0A | 0F | 14 | 19 | |
| 06 | 01 | 07 | 0D | 13 | 19 | 02 | 08 | 0E | 14 | 1A | 03 | 09 | 0F | 15 | 04 | 0A | 10 | 16 | 05 | 0B | 11 | 17 | 06 | 0C | 12 | 18 | |
| 07 | 01 | 08 | 0F | 16 | 02 | 09 | 10 | 17 | 03 | 0A | 11 | 18 | 04 | 0B | 12 | 19 | 05 | 0C | 13 | 1A | 06 | 0D | 14 | 07 | 0E | 15 | |
| 08 | 01 | 09 | 11 | 19 | 02 | 0A | 12 | 1A | 03 | 0B | 13 | 04 | 0C | 14 | 05 | 0D | 15 | 06 | 0E | 16 | 07 | 0F | 17 | 08 | 10 | 18 | |
| 09 | 01 | 0A | 13 | 02 | 0B | 14 | 03 | 0C | 15 | 04 | 0D | 16 | 05 | 0E | 17 | 06 | 0F | 18 | 07 | 10 | 19 | 08 | 11 | 1A | 09 | 12 | |
| 0A | 01 | 0B | 15 | 02 | 0C | 16 | 03 | 0D | 17 | 04 | 0E | 18 | 05 | 0F | 19 | 06 | 10 | 1A | 07 | 11 | 08 | 12 | 09 | 13 | 0A | 14 | |
| 0B | 01 | 0C | 17 | 02 | 0D | 18 | 03 | 0E | 19 | 04 | 0F | 1A | 05 | 10 | 06 | 11 | 07 | 12 | 08 | 13 | 09 | 14 | 0A | 15 | 0B | 16 | |
| 0C | 01 | 0D | 19 | 02 | 0E | 1A | 03 | 0F | 04 | 10 | 05 | 11 | 06 | 12 | 07 | 13 | 08 | 14 | 09 | 15 | 0A | 16 | 0B | 17 | 0C | 18 | |
| 0D | 01 | 0E | 02 | 0F | 03 | 10 | 04 | 11 | 05 | 12 | 06 | 13 | 07 | 14 | 08 | 15 | 09 | 16 | 0A | 17 | 0B | 18 | 0C | 19 | 0D | 1A | ← PC-9800シリーズ BASIC |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |

b）15セクタ／トラックの場合

| セクタシーケンス | 物理セクタ番号(16進) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | |
| | 論理セクタ番号(16進) | | | | | | | | | | | | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | ←IBM-PC |
| 02 | 01 | 03 | 05 | 07 | 09 | 0B | 0D | 0F | 02 | 04 | 06 | 08 | 0A | 0C | 0E | |
| 03 | 01 | 04 | 07 | 0A | 0D | 02 | 05 | 08 | 0B | 0E | 03 | 06 | 09 | 0C | 0F | |
| 04 | 01 | 05 | 09 | 0D | 02 | 06 | 0A | 0E | 03 | 07 | 0B | 0F | 04 | 08 | 0C | |
| 05 | 01 | 06 | 0B | 02 | 07 | 0C | 03 | 08 | 0D | 04 | 09 | 0E | 05 | 0A | 0F | |
| 06 | 01 | 07 | 0D | 02 | 08 | 0E | 03 | 09 | 0F | 04 | 0A | 05 | 0B | 06 | 0C | |
| 07 | 01 | 08 | 0F | 02 | 09 | 03 | 0A | 04 | 0B | 05 | 0C | 06 | 0D | 07 | 0E | |
| 08 | 01 | 09 | 02 | 0A | 03 | 0B | 04 | 0C | 05 | 0D | 06 | 0E | 07 | 0F | 08 | |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |

c）8セクタ／トラックの場合

| セクタシーケンス | 物理セクタ番号(16進) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | |
| | 論理セクタ番号(16進) | | | | | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | ←MS-DOS |
| 02 | 01 | 03 | 05 | 07 | 02 | 04 | 06 | 08 | |
| 03 | 01 | 04 | 07 | 02 | 05 | 08 | 03 | 06 | |
| 04 | 01 | 05 | 02 | 06 | 03 | 07 | 04 | 08 | |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | |

## 5.2.6 初期化(INITIALIZE)

### (1) 機能

1MB フロッピーディスクコントローラの初期化を行う(システム初期化時に1回だけ実行すればよい).

次の操作を行う.

・FDC μPD765A の初期化.

・システム共通域の初期化(DISK_EQUIP, DISK_INT, DISK_RESULT).

・FDC に対し, SPECIFY コマンドを発行(FDC の動作モードの設定).

・各装置にリキャリブレイトコマンドを発行(装置の接続状況と動作確認).

### (2) 入力

・内部割り込みコード← 1BH

・AH ←03H(BIOS コマンド識別コード)

・AL ←デバイス種別・ユニット番号(DA/UA) 9 × H

ユニット番号は指定する必要がない.

### (3) 出力

・CF ←終了条件

0 ：正常終了

1 ：異常終了

・AH ←ステータス情報(00H)

### (4) 処理

① FDC μPD765A の初期化

ライトコントロールレジスタを使用.

② システム共通域の初期化

③ FDC に対して初期値の設定(SPECIFY コマンド発行).

・HUT (Head Unload Time)の設定.

リード／ライトコマンド実行後, アンロード状態にするまでの時間を指定する. —160msec

・SRT (Step Rate Time)の設定.

シーク動作時デバイスへ送る STEP 信号の間隔時間を指定する. ——5 msec

・HLT (Head Load Time)の設定.

リード／ライトコマンドの実行開始時, ヘッドがアンロード状態であればロードさせるが, ロードした後ヘッドが安定状態になるまでの待ち時間を指定する. ——50msec

・DMA モードに設定する.

④ 各ユニットに対してリキャリブレイトコマンドによるチェックを行う.

## 5.2.7 ベリファイ(VERIFY)

### (1) 機能

指定されているデバイス種別・ユニット番号(DA/UA)のディスク装置の，指定されたセクタのデータを読み取る．ただし，メモリへの転送は行わない．

指定条件は，データの読み出し(READ DATA)コマンドとほとんど同じである．

### (2) 入力

・内部割り込みコード← 1BH

・AH ← BIOS コマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| MT | MF | r | SEEK | 0 | 0 | 0 | 1 |

他は「5.2.1 データの読み出し(READ DATA)」と同じである．参照すること．

### (3) 出力

「5.2.1 データの読み出し(READ DATA)」参照．

注：DDAM(Deleted Data Address Mark)を検出しても，そのセクタをスキップして，処理を続行する．

### (4) 処理

読み取ったデータをメモリに転送しないことを除いて「5.2.1」と同じである．

## 5.2.8 センス(SENSE)

### (1) 機能

指定したデバイスの状態を調べる．

### (2) 入力

・内部割り込みコード← 1BH

・AH ←04Hまたは84H(BIOS コマンド識別コード)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| n | 0 | r | 0 | 0 | 1 | 0 | 0 |

・AL ←デバイス種別・ユニット番号(DA/UA)(90H～93Hまたは10H～13H)

**(3) 出力**

・CF ←終了条件

　　0：正常終了

　　1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　容 |
|---|---|---|---|
| 1×H | WP | Write Protect | 媒体がセットされているが，ライトプロテクトの状態である． |
| 4×H | EC | Equipment Check | デバイスが異常か，リキャリブレイトコマンド実行時に，一定時間内にトラック0信号を確認できなかった場合の異常終了ステータス． |
| 6×H | NR | Not Ready | 媒体がセットされていない． |
| ビット0 | | | 0：片面媒体がセットされている．<br>1：両面媒体がセットされている． |
| ビット2 | | | 0：48TPI　1：96TPI　DA/UAが10H～13Hのとき通知される． |
| ビット3 | | | 0：1MBドライブ　1：1MB/640KB両用ドライブ　AHが84Hのとき通知される |

## 5.2.9 IDの読み出し(READ ID)

**(1) 機能**

指定されたデバイス種別・ユニット番号(DA/UA)のディスク装置の，指定されたトラック上の正常なIDを読み取り，IDR(C，H，R，N)に格納する．

**(2) 入力**

・内部割り込みコード← 1BH

・AH ← BIOSコマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | MF | $\bar{r}$ | SEEK | 1 | 0 | 1 | 0 |

SEEK：1：シーク動作を行う．シリンダ番号Cへシークする．／0：シーク動作を行わない．現在のシリンダ位置で操作．

$\bar{r}$：1：リトライしない．／0：8回のリトライをする．

MF：1：倍密度を使用する．／0：単密度を使用する．

・AL ←デバイス種別・ユニット番号DA/UA(90H～93H)

・CL ←シリンダ番号(C)(0～76)(AHのSEEKビットが1のとき意味をもつ)

・DH ←ヘッド番号(H)(0～1)

**(3) 出力**

・CF ←終了条件

　　0：正常終了

　　1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　　　容 |
|---|---|---|---|
| 40H | EC | Equipment Check | 「5．2．1」参照 |
| 60H | NR | Not Ready | 「5．2．1」参照 |
| C0H | ND | No Data | 1トラック分の読み取りを行っても，正しいIDがみつからない状態で異常終了． |
| E0H | MA | Missing Address Mark | 正しいIDがみつからず，しかもIDアドレスマークが1回も検出されなかった． |

・CH ←セクタ長(N)
・CL ←シリンダ番号(C)
・DH ←ヘッド番号(H)
・DL ←セクタ番号(R)
（以上 正常なIDのIDR）

**(4) 処理**

指定されたトラックで，最初に正常に読み取れたIDをIDRへ格納する．

最大1トラック分のデータを読み取ってもエラーでないIDが見つからない場合は，ステータスにNDをセットして異常終了する．

エラーでないIDが見つからず，なおかつIDアドレスマークが1回も検出されなかったときは，ステータスにMAをセットして異常終了する．

動作終了後，ヘッドをすぐにはアンロード状態にせず，一定時間経過後(約1回転＝160msec)アンロード状態にする．したがってこの時間内に以前と同じユニットに対してリードまたはライト系のコマンドが実行された場合，ヘッドロードの安定時間待ち(約50msec)を節約することが出来る．

ただし，この時間以内に他のデバイスに対しての動作が行われた場合には，以前のデバイスのヘッドはアンロード状態になる．

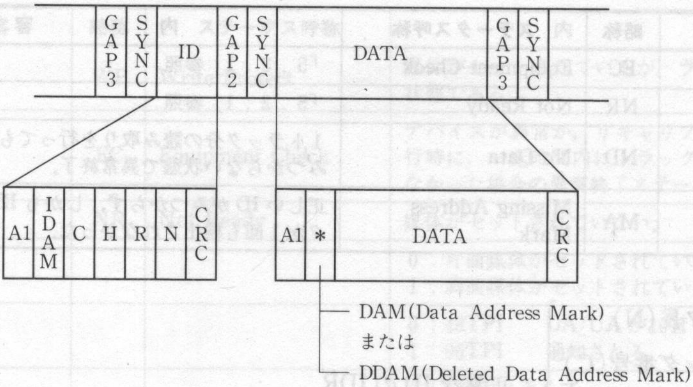
また，動作が起動された方のユニットでは自動的にロード安定時間待ちが行われる．

読み取り動作を正常終了したときにはIDRに，MTビット，および最終セクタによりIDRの状態を示す値がセットされる(5．2．1(5)参照)．

## 5.2.10 デリーテッドデータの書き込み(WRITE DELETED DATA)

### (1) 機能

セクタのデータフィールドのData Address Markの代りにDeleted Data Address Markを書き込むことを除いて，Write Dataコマンド(「5.2.2」)と同じ機能を持つ.




### (2) 入力

・内部割り込みコード←1BH

・AH←BIOSコマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| MT | MF | r | SEEK | 1 | 0 | 0 | 1 |

他は「5.2.2 Write Data」と同じ.

### (3) 出力

「5.2.2 Write Data」と同じ.

### (4) 処理

「5.2.2 Write Data」と同じ.

## 5.2.11 デリーテッドデータの読み出し(READ DELETED DATA)

(1) 機能

セクタのデータフィールドにあるData Address Markの代りにDeleted Data Address Markを扱うことを除いて，Read Dataコマンド(「5.2.1」)と同じ機能を持つ.

(2) 入力

・内部割り込みコード←1BH

・AH←BIOSコマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| MT | MF | r̄ | SEEK | 1 | 1 | 0 | 0 |

他は「5.2.1 Read Data」と同じ.

(3) 出力

「5.2.1 Read Data」と同じ.

(4) 処理

「5.2.1 Read Data」と同じ.

## 5.2.12 診断のための読み出し(READ DIAGNOSTIC)

(1) 機能

インデックスマークの直後から読み取りを開始し，IDのエラー，およびデータ部のエラーが検出されても読み取りを続行することを除いて，Read Dataコマンドと同じ機能である.

(2) 入力

・内部割り込みコード←1BH

・AH←BIOSコマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | MF | r̄ | SEEK | 0 | 0 | 1 | 0 |

他は「5.2.1 Read Data」と同じ.

(3) 出力

「5.2.1 Read Data」と同じ.

**(4) 処理**

インデックスマークの直後から読み取りを開始する．

次の相違点を除いてRead Dataコマンドと同じである．

① IDまたはデータ部のCRCエラーが検出されても読み取りを続ける．

② 読み取ったIDとIDRの比較を行うが，それらが異なっていてもステータスにND(No Data)をセットするだけで，そのセクタのデータを処理する．

③ Deleted Data Address Markのあるセクタを検出しても，コマンド実行の終了条件にはならない．

## 5.3　640KBフロッピーディスク

640KBフロッピーディスクにおいては，以下の点が1MBフロッピーディスクと異なる．

**(1) Morter on/off 制御**

約15秒の間，I/Oを実行しないと，自動的にMorter offする．

**(2) Not Readyの検出不可**

ディスクがセットされていない状態を検出できないため，I/Oを実行してもNot Readyとしてコマンドが終了しない．このため，コマンドが約2秒経過しても終了しないときは強制的にResetし，上位プログラムにはタイムアウトのステータスを通知する．

Not Readyのステータスは装置が接続されていないか，電源offのとき通知される．

### 5.3.1　データの読み出し(READ DATA)

**(1) 機能**

指定されたデバイス種別・ユニット番号(DA/UA＝70H～73H)を持つデバイスの，指定された実行開始セクタ(ID情報；シリンダ番号(C)，ヘッド番号(H)，セクタ番号(R)，セクタ長(N))から，指定されたメモリ領域(データバッファ領域の先頭アドレス)へ，指定された長さ(DTL)のデータを読み出す．

現在選択されているシリンダ位置のトラックから読み出す場合と，シーク動作を行い，トラックの選択を行ってから読み出す場合との選択が可能である．

このコマンドはシングルトラック／マルチトラックの読み出し選択指定(MT-Multi Track)および，単密度(FMモード)／倍密度(MFMモード)の読み出し選択指定(MF-Multiple Frequency modulation)が可能である．

## (2) 入力

・内部割り込みコード← 1BH

・AH ← BIOS コマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| MT | MF | r | SEEK | 0 | 1 | 1 | 0 |

$b_7$(MT)：シングルトラック (0)／マルチトラック (1)*の読み出し指定

（＊同一シリンダの両面トラックのみ）

$b_6$(MF)：単密度 (0)／倍密度 (1) の読み出し指定

$b_5$(r)：8回リトライする (0)／リトライなし (1) の指定

$b_4$(SEEK)：シーク動作が必要 (1)／現在のトラック位置 (0) からの読み出し指定

・AL ←デバイス種別・ユニット番号(DA/UA)(70H～73H)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | x2 | x1 |

x2 x1：接続ディスク装置のドライブに対応（00←0～11←3）

・BX ←データ長(DTL)(バイト単位)

・CL ←シリンダ番号(C)(0～79)

・DH ←ヘッド番号(H)(0～1)

・DL ←セクタ番号(R)(1～16, 9, 5)

・CH ←セクタ長(N)(00H～03H)

（CL・DH・DL・CH：これらをあわせて ID 情報と呼ぶ(略称 IDR)）

| CH | 00 | 01 | 02 | 03 |
|---|---|---|---|---|
| セクタ長(バイト／セクタ) | 128 | 256 | 512 | 1024 |

・ES/BP ←データバッファ領域の先頭アドレス

## (3) 出力

・CF ←出力条件

0：正常終了

1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　容 |
|---|---|---|---|
| 0 0 0 1××××<br>(10H) | CM | Control Mark | DDAM (Deleted Data Address Mark)を検出した．そのセクタを読み出した後，正常終了する． |
| 0 0 1 0××××<br>(20H) | DB | DMA Boundary | メモリアドレスがバンクにまたがるか，奇数番地から始まるように指定した場合の異常終了ステータス(使われていない)． |
| 0 0 1 1××××<br>(30H) | EN | End of Cylinder | 1回の動作の転送容量を超えてデータ長(DTL)を指定した場合の異常終了ステータス． |
| 0 1 0 0××××<br>(40H) | EC | Equipment Check | デバイスが異常か，リキャリブレイトコマンド実行時に，一定時間内にトラック0信号を確認できなかった場合の異常終了ステータス． |
| 0 1 0 1××××<br>(50H) | OR | Over Run | セクタのデータをメモリに転送しているとき，一定時間内にデータ転送が終了できなかった場合の異常終了ステータス． |
| 0 1 1 0××××<br>(60H) | NR | Not Ready | ユニットがノットレディ状態の場合の異常終了ステータス． |
| 1 0 1 0××××<br>(A0H) | DE | Data Error | ID読み出し時にCRCエラーが発生した場合の異常終了ステータス． |
| 1 1 0 0××××<br>(C0H) | ND | No Data | トラック内に，指定されたセクタが見つからなかった場合の異常終了ステータス． |
| 1 1 1 0××××<br>(E0H) | MA | Missing Address mark | トラック内に，指定されたセクタが見つからず，かつIDが1個もなかった場合の異常終了ステータス．(注) |

注：指定されたセクタのIDを検出した後，データ部のアドレスマークを一定時間以内に検出できなかった場合にもMAで異常終了する．

・システム共通域(F2DD RESULT 5D0～5DFの16バイトエントリ)
　　← FDCからのリザルトステータス情報を格納

**(4) 処理**

① DA/UAに対応するシステム共通域F2DD_RESULTの内容を調べる．

② SEEKビットが1(オン)かどうかを判定する．
オンならばシリンダ番号Cによってシーク動作を行う．シーク動作が終了したか，またはSEEKビットが0(オフ)ならば③の処理を行う．

③ DMAを起動する．
BX，ES，BPを送る．

④ FDCにREADコマンドを発行する．
マルチトラックMT，MFMモードはBIOSコマンドに従う．DDAM (Deleted Data Address Mark)を検出した場合は，そのセクタの読み取り後，コマンドの実行を正常終了する．

⑤ FDCに対して，READコマンドのパラメータを順次送出する．
C(シリンダ#)，H(ヘッド#)，R(セクタ#)，N(セクタ長)

⑥ 割り込み信号の入力を待つ.

入力処理の終了を待ち，リザルトステータス情報を読み出し(ST0, ST1, ST2)，実行終了セクタのID情報(C, H, R, N)の読み出しを行い，システム共通域のRESULT_DATA, BIOSコマンド出力条件件の設定を行う.

⑦ DMAチャネルを閉じる.

## (5) その他

a) 1回の動作の転送容量はMT(マルチトラック)ビット，MF(倍密度指定)ビット，N(セクタ長)に関係する.

| MT | MF | N | 転送容量(バイト) | | 最終セクタ |
|---|---|---|---|---|---|
| 0 | 0 | 00H | 1～128×n | n＝1～16 | ヘッド0のセクタ16またはヘッド1のセクタ16 |
| | 1 | 01H | 1～256×n | | |
| 1 | 0 | 00H | 1～128×n | n＝1～32 | ヘッド1のセクタ16 |
| | 1 | 01H | 1～256×n | | |
| 0 | 0 | 01H | 1～256×n | n＝1～9 | ヘッド0のセクタ9またはヘッド1のセクタ9 |
| | 1 | 02H | 1～512×n | | |
| 1 | 0 | 01H | 1～256×n | n＝1～18 | ヘッド1のセクタ9 |
| | 1 | 02H | 1～512×n | | |
| 0 | 0 | 02H | 1～512×n | n＝1～5 | ヘッド0のセクタ5またはヘッド1のセクタ5 |
| | 1 | 03H | 1～1024×n | | |
| 1 | 0 | 02H | 1～512×n | n＝1～10 | ヘッド1のセクタ5 |
| | 1 | 03H | 1～1024×n | | |

注：nの値はプログラマブルで，EOTの値で定義される.

b）BIOSコマンド実行を正常に終了したときに，システム共通域のRESULT_DATAに設定されるIDR(C，H，R，N)の状態は次表のようになる．

| MT | EOT | 最終バイトの転送に関係したセクタ | IDRの状態 C | H | R | N |
|---|---|---|---|---|---|---|
| 0 | 10H<br>09H<br>05H | ヘッド0のセクタ1～15<br>〃　1～8<br>〃　1～4 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 10H<br>09H<br>05H | ヘッド0のセクタ16<br>〃　9<br>〃　5 | C＋1 | もとのまま | R＝01[注] | もとのまま |
| | 10H<br>09H<br>05H | ヘッド1のセクタ1～15<br>〃　1～8<br>〃　1～4 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 10H<br>09H<br>05H | ヘッド1のセクタ16<br>〃　9<br>〃　5 | C＋1 | もとのまま | R＝01[注] | もとのまま |
| 1 | 10H<br>09H<br>05H | ヘッド0のセクタ1～15<br>〃　1～8<br>〃　1～4 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 10H<br>09H<br>05H | ヘッド0のセクタ16<br>〃　9<br>〃　5 | もとのまま | 下1ビットを反転する | R＝01[注] | もとのまま |
| | 10H<br>09H<br>05H | ヘッド1のセクタ1～15<br>〃　1～8<br>〃　1～4 | もとのまま | もとのまま | R＋1 | もとのまま |
| | 10H<br>09H<br>05H | ヘッド1のセクタ16<br>〃　9<br>〃　5 | C＋1 | 下1ビットを反転する | R＝01[注] | もとのまま |

注：ヘッドの選択状態(ST0のHDビットも含む)は最終バイトの転送に関係したセクタに属するトラックのままである．

c）動作終了後，ヘッドをすぐにはアンロード状態にせず，一定時間経過後(約1回転＝200msec)アンロード状態にする．したがってこの時間内に以前と同じユニットに対してリードまたはライト系のコマンドが実行された場合，ヘッドロードの安定時間待ち(約50msec)を節約することが出来る．

ただし，この時間以内に他のデバイスに対しての動作が行われた場合には，以前のデバイスのヘッドはアンロード状態になる．

また，動作が起動された方のユニットでは自動的にロード安定時間待ちが行われる．

## 5.3.2　データの書き込み(WRITE DATA)

### (1) 機能

指定されたデバイス種別·ユニット番号(DA/UA＝70H～73H)を持つデバイスの，指定された実行開始セクタ(ID情報；シリンダ番号(C)，ヘッド番号(H)，セクタ番号(R)，セクタ長(N))から，指定されたメモリ領域(データバッファ領域の先頭アドレス)の，指定された長さ(DTL)のデータを書き込む.

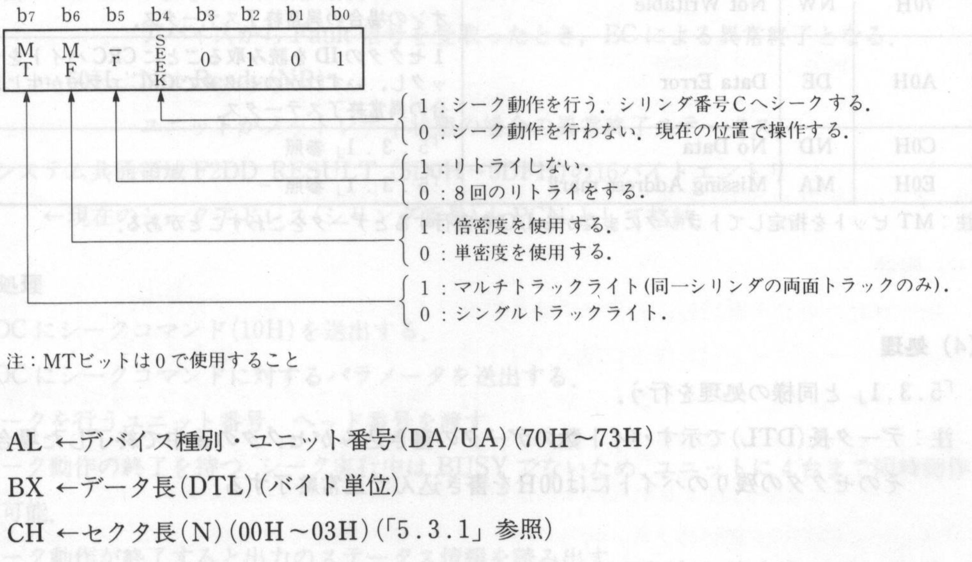
現在選択されているシリンダ位置のトラックへ書き込む場合と，シーク動作を行い，トラックの選択を行ってから書き込む場合との選択が可能である.また,単密度(FMモード)／倍密度(MFMモード)の書き込み選択指定(MF-Multiple Frequency modulation)も可能である.

注：マルチトラック指定は使用しないこと.

### (2) 入力

・内部割り込みコード←1BH

・AH←BIOSコマンド識別コード




注：MTビットは0で使用すること

・AL ←デバイス種別・ユニット番号(DA/UA)(70H～73H)

・BX ←データ長(DTL)(バイト単位)

・CH ←セクタ長(N)(00H～03H)(「5.3.1」参照)

・CL ←シリンダ番号(C)(0～79)

・DH ←ヘッド番号(H)(0～1)

・DL ←セクタ番号(R)(1～16，9，5)

・ES/BP←データバッファ領域の先頭アドレス

**(3) 出力**

・CF ←終了条件

　0：正常終了

　1：異常終了

・AH ←ステータス情報(次表参照)

・システム共通域(F2DD_RESULT 5D0H～5DFH の16バイトエントリ)

　←FDC からのリザルトステータス情報を格納

| AH の内容 | 略称 | ステータス呼称 | 内容 |
|---|---|---|---|
| 30H | EN | End of Cylinder | 「5．3．1」参照 |
| 40H | EC | Equipment Check | データの書き込み時，デバイスの FAULT 状態を検出した場合の異常終了ステータス． |
| 50H | OR | Over Run | セクタのデータ部にメモリからデータ転送をしているとき，一定時間内にデータを転送できなかった場合の異常終了ステータス．そのセクタを書き込み後，OR となる． |
| 60H | NR | Not Ready | 「5．3．1」参照 |
| 70H | NW | Not Writable | コマンド実行開始時，WRITE PROTECTED 信号がオンの場合の異常終了ステータス． |
| A0H | DE | Data Error | 1セクタの ID を読み取るごとに CRC バイトをチェックし，いずれかのセクタで CRC エラーが生じた場合の異常終了ステータス． |
| C0H | ND | No Data | 「5．3．1」参照 |
| E0H | MA | Missing Address mark | 「5．3．1」参照 |

注：MT ビットを指定してトラックにまたがる I/O を実行するとデータをこわすことがある．

**(4) 処理**

「5．3．1」と同様の処理を行う．

注：データ長(DTL)で示すバイト数のデータの書き込みがセクタの途中で終了した場合は，そのセクタの残りのバイトには00H を書き込んで正常終了する．

## 5.3.3　シークを行う(SEEK)

**(1) 機能**

指定されたデバイス種別・ユニット番号(DA/UA)に対して，指定されたシリンダ物理番号(C)までシークさせる．

なお，読み出し(READ)／書き込み(WRITE)用の BIOS コマンドには，シーク動作を同時に行うことが可能になっている．それぞれの BIOS コマンド識別コードの $b_4$ ビット(10H)をオン(1) にするとシーク動作を伴い，オフ (0) にすると現在のシリンダを対象とした動作を行う．

(2) 入力

・内部割り込みコード← 1BH

・AH ←10H(BIOS コマンド識別コード)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | × | $\overline{r}$ | 1 | × | × | × | × |

・AL ←シークを行うデバイス種別・ユニット番号(DA/UA)(70H～73H)

・CL ←シークを行うシリンダ番号(C)(0～79)

(3) 出力

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

40H：Equipment Check(EC)

デバイスから Fault 信号を受取ったとき，EC による異常終了となる.

60H：Not Ready(NR)

ユニットがノットレディ状態の場合の異常終了ステータス.

・システム共通領域 F2DD_RESULT (5D0H～5DFH)の16バイトエントリ

←現在のシークアドレス(シリンダ番号)を PCN として格納.

(4) 処理

① FDC にシークコマンド(10H)を送出する.

② FDC にシークコマンドに対するパラメータを送出する.

シークを行うユニット番号，ヘッド番号を渡す.

③ シーク動作の終了を待つ. シーク実行中は BUSY でないため、ユニットに4台まで同時動作が可能.

④ シーク動作が終了すると出力のステータス情報を読み出す.

### 5.3.4 シリンダ0へシークする(RECALIBRATE―リキャリブレイト)

(1) 機能

シリンダ物理番号0へ(デバイスから，トラック0信号を検出するまで)シークさせる.

シーク動作は，シリンダ物理番号0の方向へ1シリンダずつ行い，デバイスからのトラック0信号を検出するまでくり返す.

**(2) 入力**

・内部割り込みコード← 1BH

・AH ← 07H(BIOSコマンド識別コード)

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $\overline{r}$ | 0 | 0 | 1 | 1 | 1 |

・AL ←デバイス種別・ユニット番号(DA/UA)(70H～73H)

**(3) 出力**

・CF ←終了条件

　0：正常終了

　1：異常終了

・AH ←ステータス情報

　40H：Equipment Check(EC)

　　デバイスからのトラック0信号がオン (1) になるまで154回くり返すが，検出できなかった場合，ECによる異常終了となる．

　60H：Not Ready(NR)

　　ユニットがノットレディ状態の場合の異常終了ステータス．

## 5.3.5　トラックのフォーマット(FORMAT TRACK)

**(1) 機能**

指定されたセクタ長(N)，トラック当りのセクタ数(SC)，ギャップ長(GPL)，データ部に書き込むデータパターン(D)などに従って1トラック分フォーマット書き込みを行う．

セクタのID部にデータを書き込むときには，指定されたデータバッファ領域の内容(C，H，R，Nの4バイト×セクタ数)を使用する．したがって，セクタシーケンスIDや不良シリンダIDなども書き込むことが可能である(C，H，R，Nの指定による)．

各セクタのデータ部には，1バイトのデータDをNで指定される長さ分だけくり返し書き込む．

(2) 入力

・内部割り込みコード←1BH

・AH ← BIOS コマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| 0 | MF | $\overline{r}$ | SEEK | 1 | 1 | 0 | 1 |

SEEK（b4）：
- 1：シーク動作を行う．シリンダ番号Cへシークする．
- 0：シーク動作を行わない．現在のシリンダ位置で操作する．

$\overline{r}$（b5）：
- 1：リトライしない．
- 0：8回のリトライをする．

MF（b6）：
- 1：倍密度を使用する．
- 0：単密度を使用する．

・AL ←デバイス種別・ユニット番号(DA/UA)(70H～73H)

・BX ←データ長(DTL)(バイト単位)

・CH ←セクタ長(N)(00, 01, 02, 03)

・CL ←シリンダ番号(C)(0～79)(AH の SEEK ビットが1のとき有効)

・DH ←ヘッド番号(H)(0～1)

・DL ←データ部への書き込みデータパターン(D)

・ES/BP ←データバッファ領域の先頭アドレス

(3) 出力

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　容 |
|---|---|---|---|
| 40H | EC | Equipment Check | 書き込み終了時に Fault 状態を検出した. |
| 50H | OR | Over Run | セクタの ID 部に書き込み時, 一定時間内にデータ転送ができなかった. |
| 60H | NR | Not Ready | デバイスがノットレディ状態になった. |
| 70H | NW | Not Writable | 「5.3.1」参照 |

### (4) 処理

① DA/UAで指定されるユニットの，現在選択されているシリンダのHバイトで示されるヘッドの面にあるトラックにフォーマットを書き込む．

② セクタのID部の書き込みは，指定されているデータバッファ領域の先頭アドレス(ES : BP)から，BXのデータ長までのデータバッファ上に，セクタのID部ごとのC，H，R，Nの4バイトエントリをトラックのセクタ数分だけ展開しておき，これを書き込む((5) を参照)．

③ セクタのデータ部の書き込みは，指定されたDL(データパターンD)の内容を，指定されたCH(セクタ長N)の長さ分だけセクタごとに繰り返す．

### (5) セクタのID部をフォーマットするための準備

① パラメータの指定方法

| 指定方法 | | 指定内容 | | トラック当りのセクタ数 SC | ギャップ長 GPL | IDフォーマット用データバッファ長 (DTL) | 備考 |
|---|---|---|---|---|---|---|---|
| MF ビット | N セクタ長 | 密度 | | | | | |
| 0 | 00 | 単密度 | 128 | 10H (16) | 1BH | 64 | |
| | 01 | FM モード | 256 | 09H (9) | 2AH | 36 | |
| | 02 | | 512 | 05H (5) | 3AH | 20 | |
| 1 | 01 | 倍密度 | 256 | 10H (16) | 33H | 64 | PC-9800シリーズ $N_{88}$-BASIC |
| | 02 | MFM モード | 512 | 09H (9) | 50H | 36 | IBMの8セクタ／トラックとするときは32バイトにする |
| | 03 | | 1024 | 05H (5) | 74H | 20 | |

注：DTLは上表の値以上であればよい．256をとるようにする．

②セクタシーケンスに従った論理セクタ番号

a）16セクタ／トラックの場合

| セクタシーケンス | 物理セクタ番号(16進) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0E | 0C | 0D | 0E | 0F | 10 |
| | 論理セクタ番号(16進) | | | | | | | | | | | | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 |
| 02 | 01 | 03 | 05 | 07 | 09 | 0B | 0D | 0F | 02 | 04 | 06 | 08 | 0A | 0C | 0E | 10 |
| 03 | 01 | 04 | 07 | 0A | 0D | 10 | 02 | 05 | 08 | 0B | 0E | 03 | 06 | 09 | 0C | 0F |
| 04 | 01 | 05 | 09 | 0D | 02 | 06 | 0A | 0F | 03 | 07 | 0B | 0F | 04 | 08 | 0C | 10 |
| 05 | 01 | 06 | 0B | 10 | 02 | 07 | 0C | 03 | 08 | 0D | 04 | 09 | 0E | 05 | 0A | 0F |
| 06 | 01 | 07 | 0D | 02 | 08 | 0E | 03 | 09 | 0F | 04 | 0A | 10 | 05 | 0B | 06 | 0C |
| 07 | 01 | 08 | 0F | 02 | 09 | 10 | 03 | 0A | 04 | 0B | 05 | 0C | 06 | 0D | 07 | 0E |
| 08 | 01 | 09 | 02 | 0A | 03 | 0B | 04 | 0C | 05 | 0D | 06 | 0E | 07 | 0F | 08 | 10 |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

b）9セクタ／トラックの場合

| セクタシーケンス | 物理セクタ番号(16進) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
| | 論理セクタ番号(16進) | | | | | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
| 02 | 01 | 03 | 05 | 07 | 09 | 02 | 04 | 06 | 08 |
| 03 | 01 | 04 | 07 | 02 | 05 | 08 | 03 | 06 | 09 |
| 04 | 01 | 05 | 09 | 02 | 06 | 03 | 07 | 04 | 08 |
| 05 | 01 | 06 | 02 | 07 | 03 | 08 | 04 | 09 | 05 |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

c）8セクタ／トラックの場合

| セクタシーケンス | 物理セクタ番号(16進) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| | 論理セクタ番号(16進) | | | | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| 02 | 01 | 03 | 05 | 07 | 02 | 04 | 06 | 08 |
| 03 | 01 | 04 | 07 | 02 | 05 | 08 | 03 | 06 |
| 04 | 01 | 05 | 02 | 06 | 03 | 07 | 04 | 08 |
| FF | FF | FF | FF | FF | FF | FF | FF | FF |

d）5セクタ／トラックの場合

| セクタ シーケンス | 物理セクタ番号 | | | | |
| --- | --- | --- | --- | --- | --- |
| | 01 | 02 | 03 | 04 | 05 |
| | 論理セクタ番号 | | | | |
| 00,01 | 01 | 02 | 03 | 04 | 05 |
| 02 | 01 | 03 | 05 | 02 | 04 |
| 03 | 01 | 04 | 02 | 05 | 03 |
| FF | FF | FF | FF | FF | FF |

### (6) 例

① DLに格納するデータパターン(D)について

・BASICの場合　：40H(PC-9800シリーズ)

・CP/M-86の場合：E5H

・MS-DOSの場合　：E5H

② データバッファの大きさ

・26セクタ／トラックの場合：4バイト×26

・8セクタ／トラックの場合：4バイト×8

## 5.3.6　初期化(INITIALIZE)

### (1) 機能

640KBフロッピーディスクコントローラの初期化を行う．システム初期化時に1回だけ実行すればよい．

次の操作を行う．

・FDC μPD765Aの初期化．

・システム共通域の初期化(DISK_EQUIP, DISK_INT, F2DD_RESULT)．

・FDCに対し，SPECIFYコマンドを発行(FDCの動作モードの設定)

・各装置にリキャリブレイトコマンドを発行(装置の接続状況と動作確認)．

### (2) 入力

内部割り込みコード←1BH

・AH ←03H(BIOSコマンド識別コード)

・AL ←デバイス種別・ユニット番号(DA/UA)(7XH)

　　ユニット番号は指定する必要がない

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報(00H)

**(4) 処理**

① FDC μPD765A の初期化

ライトコントロールレジスタを使用.

② システム共通域の初期化

③ FDC に対して初期値の設定(SPECIFY コマンド発行).

・HUT (Head Unload Time)の設定.

リード/ライトコマンド実行後，アンロード状態にするまでの時間を指定する——192msec

・SRT (Step Rate Time)の設定.

シーク動作時デバイスへ送る STEP 信号の間隔時間を指定する——4msec

・HLT (Head Load Time)の設定.

リード/ライトコマンドの実行開始時，ヘッドがアンロード状態であればロードさせるが，ロードした後ヘッドが安定状態になるまでの待ち時間を指定する——50msec

・DMA モードに設定する.

④ 各ユニットに対してリキャリブレイトコマンドによるチェックを行い，トラック0を検出したユニットの，対応する Equipment フラグを ON にする.

⑤ 各ユニットのオペレーションモードを 2DD モードにする.

## 5.3.7 ベリファイ(VERIFY)

**(1) 機能**

指定されているデバイス種別・ユニット番号(DA/UA)のディスク装置の，指定されたセクタのデータを読み取る．ただし，メモリへ転送は行わない.

指定条件は，データの読み出し(READ DATA)コマンドとほとんど同じである.

**(2) 入力**

・内部割り込みコード← 1BH

・AH ← BIOS コマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| MT | MF | $\bar{r}$ | SEEK | 0 | 0 | 0 | 1 |

他は「5.3.1データの読み出し(READ DATA)」と同じである．参照すること．

**(3) 出力**

「5.3.1データの読み出し(READ DATA)参照．

注：DDAM (Deleted Data Address Mark)を検出しても，そのセクタをスキップして，処理を続行する．

**(4) 処理**

読み取ったデータをメモリに転送しないことを除いて「5.3.1」と同じである．

### 5.3.8　センス(SENSE)

**(1) 機能**

指定したデバイスの状態を調べる．

**(2) 入力**

・内部割り込みコード←1BH

・AH ←04H(BIOS コマンド識別コード)

・AL ←デバイス種別・ユニット番号(DA/UA)(70H～73H)

**(3) 出力**

・CF ←終了条件

　　0：正常終了(ビット0～3有効)

　　1：異常終了(ビット0～3無効)

・AH ←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　　容 |
|---|---|---|---|
| 1×H | WP | Write Protect | 媒体がセットされているが，ライトプロテクトの状態である． |
| 4×H | EC | Equipment Check | 「5．3．1」参照 |
| 6×H | NR | Not Ready | 媒体がセットされていない． |
| ビット0 | | | 0：片面モード<br>1：両面モード |
| ビット2 | | | 0：40シリンダモード<br>1：80シリンダモード |

注：Set Operation Mode コマンドで設定した値が UNIT 毎に通知される．

## 5.3.9 IDの読み出し(READ ID)

### (1) 機能

指定されたデバイス種別·ユニット番号(DA/UA)のディスク装置の，指定されたトラック上の正常なIDを読み取り，IDR (C, H, R, N)に格納する.

### (2) 入力

・内部割り込みコード← 1BH

・AH ← BIOSコマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| 0 | MF | $\overline{\text{r}}$ | SEEK | 1 | 0 | 1 | 0 |

SEEK（$b_4$）
1：シーク動作を行う．シリンダ番号Cへシークする.
0：シーク動作を行わない．現在のシリンダ位置で操作する.

$\overline{\text{r}}$（$b_5$）
1：リトライしない.
0：8回のリトライをする.

MF（$b_6$）
1：倍密度を使用する.
0：単密度を使用する.

・AL ←デバイス種別・ユニット番号(DA/UA)(70H~73H)

・CL ←シリンダ番号(C)(0~76)(AHのSEEKビットが1のとき意味をもつ)

・DH ←ヘッド番号(H)(0～1)

### (3) 出力

・CF ←終了条件

0：正常終了

1：異常終了

・AH←ステータス情報

| AHの内容 | 略称 | ステータス呼称 | 内　　容 |
|---|---|---|---|
| 40H | EC | Equipment Check | 「5．3．1」参照 |
| 60H | NR | Not Ready | 「5．3．1」参照 |
| C0H | ND | No Data | 1トラック分の読み取りを行っても，正しいIDがみつからない状態で異常終了. |
| E0H | MA | Missing Address Mark | 正しいIDがみつからず，しかもIDアドレスマークが1回も検出されなかった. |

・CH ←セクタ長(N)
・CL ←シリンダ番号(C)
・DH ←ヘッド番号(H)
・DL ←セクタ番号(R)

（以上4項目）正常なIDのIDR

**(4) 処理**

指定されたトラックで，最初に正常に読み取れたIDをIDRへ格納する．

最大1トラック分のデータを読み取ってもエラーでないIDが見つからない場合は，ステータスにNDをセットして異常終了する．

エラーでないIDが見つからず，なおかつIDアドレスマークが1回も検出されなかったときは，ステータスにMAをセットして異常終了する．

動作終了後，ヘッドをすぐにはアンロード状態にせず，一定時間経過後(約1回転＝160msec)アンロード状態にする．したがってこの時間内に以前と同じユニットに対してリードまたはライト系のコマンドが実行された場合，ヘッドロードの安定時間待ち(約50msec)を節約することが出来る．

ただし，この時間以内に他のデバイスに対しての動作が行われた場合には，以前のデバイスのヘッドはアンロード状態になる．

また，動作が起動された方のユニットでは自動的にロード安定時間待ちが行われる．

読み取り動作を正常終了したときにはIDRに，MTビット，および最終セクタによりIDRの状態を示す値がセットされる(5.2.1(5)参照)．

## 5.3.10　オペレーションモードの設定

**(1) 機能**

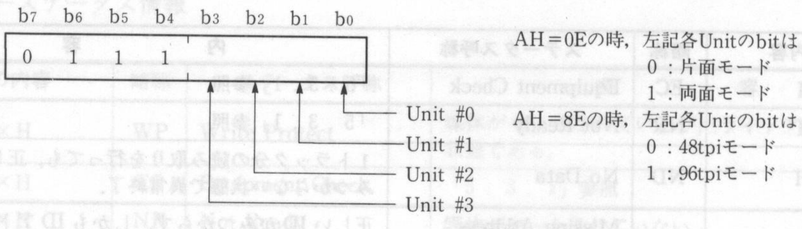
640KB FDのオペレーションモードを指定する．

**(2) 入力**

・内部割り込みコード← 1BH

・AH ← 0EH または 8EH

・AL ← 1×H




**(3) 出力**

・CF ← 0
・AH ← 0
} 正常終了

# 5.4　固定ディスク

**●相対アドレスによるアクセス**

固定ディスクにアクセスする場合，ボリューム上の相対セクタアドレスによってアクセスすることができる．使い方は次のとおり．

① デバイス種別・ユニット番号(DA/UA)←00H～01H

② 相対アドレスの指定

| DH | DL | CH | CL |
|---|---|---|---|
| × × × × × × × × | 0 0 0 $d_4$ $d_3$ $d_2$ $d_1$ $d_0$ | $d_7$ $d_6$ $d_5$ $d_4$ $d_3$ $d_2$ $d_1$ $d_0$ | $d_7$ $d_6$ $d_5$ $d_4$ $d_3$ $d_2$ $d_1$ $d_0$ |

MSB　　相対セクタアドレスS　　LSB

③ 相対セクタアドレスと絶対セクタアドレスとの関係

相対セクタアドレスS＝トラックあたりのセクタ数(ヘッド数$x_3$＋$x_2$)＋$x_1$

絶対セクタアドレス($x_3$，$x_2$，$x_1$)

シリング番号$x_3$：(0～152または0～309または0～307または0～614)

ヘッド番号　$x_2$：(0～3または0～7)

セクタ番号　$x_1$：(0～32または0～16)

④ IDRを指定するコマンドに適応される．

## 5.4.1　データの読み出し(READ DATA)

### (1) 機能

指定されたデバイスの指定されたデイスクアドレス(セクタ)から，指定された長さのデータを，メモリ上の指定されたデータバッファへ読み出す．

### (2) 入力

・内部割り込みコード←01BH

・AH←BIOSコマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| × | × | $\bar{r}$ | × | 0 | 1 | 1 | 0 |

リトライ指定（$b_5$）
- 0：エラー発生時8回のリトライを行う．
- 1：リトライをしない．
- エラーの種類に応じてリキャリブレイトも行う．

・AL ←デバイス種別・ユニット番号(DA/UA)(80H～81H)

・BX ←データの長さ

256×n：PC-9801-27

512×n：XL/XA 内蔵インターフェイス

・CX ←ディスクアドレス(シリンダ番号)

5MB ：0～152

10MB ：0～309

20MB ：0～307(厚型)

：0～614(薄型)

40MB ：0～614

・DH ←ディスクアドレス(ヘッド番号)

5,10MB ：0～3

20MB ：0～7(厚型)

：0～3(薄型)

40MB ：0～7

・DL ←ディスクアドレス(セクタ番号)

0～32 ：PC-9801-27

0～16 ：XL/XA 内蔵インターフェイス

・ES/BP ←バッファアドレス

バッファの先頭アドレスを示す

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

DB, EN, EC, OR, NR, DE, ND, MA, CM

**(4) 処理**

ALで指定されたデバイス種別・ユニット番号(DA/UA)のデバイス上の，指定されたディスクアドレスIDR (CX, DH, DL)から，指定されたデータ長(BX)のデータを読み取り，主記憶上の指定されたデータバッファ(先頭アドレスES：BPからBXで指定された長さ)へ転送する．ハードウェア上の動作は次のようになる．

①1セクタ分のデータ転送終了後，指定されたセクタ分のデータが転送されていなければ，次のセクタを処理する．

②指定されたセクタ数のデータが，トラック，またはシリンダにまたがっていた場合，自動

的にトラック，またはシリンダを切り替えて処理を続ける．

③指定されたセクタ数だけのデータを転送したら正常終了する．

コマンド実行中にエラーを検出したときは，その内容をステータス情報としてAHにセットし，異常終了する．ただし，BIOSコマンド識別コードの$\bar{r}$ビットがゼロのとき，エラーを通知するまでに8回の再試行を実行する(このリトライ機能はハードウェアで行うもので，ソフトウェアによるものではない)．

また，エラーの種類に応じてリキャリブレイト，再シークも行われ，もし成功すればエラーを通知せず正常終了する．

## 5.4.2　データの書き込み(WRITE DATA)

### (1) 機能

メモリ上の指定されたデータバッファから，指定されたデバイスの指定されたディスクアドレス(セクタ)へ，指定された長さのデータを書き込む．

### (2) 入力

・内部割り込みコード← 01BH
・AH ← BIOS コマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| × | × | $\bar{r}$ | × | 0 | 1 | 0 | 1 |

$\bar{r}$：リトライ指定　0：エラー発生時8回のリトライを行う．
　　　　　　　　　　1：リトライをしない．

・AL ←デバイス種別・ユニット番号(DA/UA)(80H～81H)

・BX ←データの長さ

　256×n：PC-9801-27
　512×n：XL/XA 内蔵インターフェイス

・CX ←ディスクアドレス(シリンダ番号)

　5MB ：0～152
　10MB ：0～309
　20MB ：0～307(厚型)
　　　 ：0～614(薄型)
　40MB ：0～614

・DH ←ディスクアドレス(ヘッド番号)

　5, 10MB：0～3
　20MB　：0～7(厚型)
　　　　：0～3(薄型)
　40MB　：0～7

・DL ←ディスクアドレス(セクタ番号)

　0～32：PC-9801-27

　0～16：XL/XA 内蔵インターフェイス

・ES/BP ←メモリ上のバッファアドレス

　バッファの先頭アドレスを示す

(3) 出力

・CF ←終了条件

　0：正常終了

　1：異常終了

・AH ←ステータス情報

　DB, EN, EC, OR, NR, DE, ND, MA, NW

(4) 処理

メモリ上の指定されたデータバッファの先頭アドレス(ES:BP)から，指定された長さ(BX)のデータを，ALで指定したデバイスのディスクアドレスIDR(CX, DH, DL)へ書き込む．

ハードウェア上の動作，エラー検出等の制御は「5.4.1」と同様である．メモリからディスクへの書き込み動作である点だけが異なる．

### 5.4.3 リキャリブレイト(RECALIBRATE)

(1) 機能

指定されたデバイスのアームをシリンダ0へシークさせる．

(2) 入力

・内部割り込みコード← 01BH

・AH ← BIOS コマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| × | × | $\bar{r}$ | × | 0 | 1 | 1 | 1 |

・AL ←デバイス種別・ユニット番号(DA/UA)(80H～81H)

(3) 出力

・CF ←終了条件

　0：正常終了

　1：異常終了

・AH ←ステータス情報

　EC, NR

**(4) 処理**

AHで指定されたデバイスのアームを物理シリンダ0へ移動する.

エラーリトライ制御は「5.4.1」と同様である.

## 5.4.4 リトラクト(RETRACT)

**(1) 機能**

デバイス種別・ユニット番号(DA/UA)で指定されるデバイスに対して，アームを未使用シリンダへ移動する.

**(2) 入力条件**

・内部割り込みコード← 01BH

・AH ← BIOSコマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| × | × | $\bar{r}$ | × | 1 | 1 | 1 | 1 |

・AL ←デバイス種別・ユニット番号(DA/UA)(80H～81H)

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

**(4) 処理**

デバイス種別・ユニット番号(DA/UA)で指定されるデバイスのアームを，意味のないシリンダ位置に移動する(データを書き込まない「未使用シリンダ」が設けられている).

コマンド実行中にエラーを検出したときは，そのエラーステータス情報をAHにセットし，異常終了する．ただし，コマンド識別コードの$\bar{r}$ビット($b_5$)が0のときは，エラーを通知する前に8回の再試行を実行する．もし，再試行が成功すれば，エラーを通知せずに正常終了する.

**(5) 使用上の注意**

電源をオフにする前には必ずリトラクトコマンドを発行し，アームを未使用シリンダに位置づけなければならない．これは電源をオンにした時にアームが記録面を傷つけないようにするためである.

### 5.4.5　IDの書き込み(FORMAT TRACK/DRIVE)

**(1) 機能**

デバイス種別・ユニット番号(DA/UA)で指定されるデバイスに対して，トラック単位，またはデバイス単位にセクタフォーマットを行う．

ID部にはインタリーブファクタに従ったセクタ番号が書き込まれ，データ部にはE5Hが書き込まれる．

**(2) 入力**

・内部割り込みコード← 01BH

・AH ← BIOSコマンド識別コード

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| d | × | $\bar{r}$ | × | 1 | 1 | 0 | 1 |

d：フォーマット単位の指定　0：トラック単位．　1：デバイス単位．

・AL ←デバイス種別・ユニット番号(DA/UA)(80H～81H)

・BH ←インタリーブファクタ(1～16)(注1)

・CX ←トラック単位にフォーマットする場合のシリンダ番号(注2)

・DH ←トラック単位にフォーマットする場合のヘッド番号(注2)

・DL ← 0

注1：インタリーブファクタは通常5を指定する．

注2：デバイス単位にフォーマットする場合にはCX ← 0，DH ← 0とする．

**(3) 出力**

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報

DB, EC, OR, NR, NW

**(4) 処理**

正常処理についての動作は「(1) 機能」のとおり．

コマンド実行中にエラーを検出したときは，その内容をステータス情報としてAHに格納して異常終了する．ただし，BIOSコマンド識別コードの$\bar{r}$(入力条件AHの$b_5$)が0のとき，すなわちリトライ要求がある場合には，エラーを通知する前に8回の再試行が行われる．エラーの種類に応じてリキャリブレイト，再シークが行われる．なお，再試行が成功すればエラーを通知せず正常終了する．このリトライ処理はハードウェアで行うものである．

### (5) インタリーブファクタについて

指定されたフォーマット単位によって，それぞれのトラックをセクタごとにフォーマットする．

ID部のセクタアドレス(シリンダ番号，ヘッド番号，セクタ番号)にはインタリーブファクタに従った論理セクタ番号が書き込まれる．物理的なセクタシーケンスとID部に書き込まれる論理セクタ番号は次のような関係になる．

| インタリー | 物理セクタ番号 (16進) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ブファクタ | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | 20 |
| (16進) | 論理セクタ番号 (16進) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | 20 |
| 02 | 00 | 02 | 04 | 06 | 08 | 0A | 0C | 0E | 10 | 12 | 14 | 16 | 18 | 1A | 1C | 1E | 20 | 01 | 03 | 05 | 07 | 09 | 0B | 0D | 0F | 11 | 13 | 15 | 17 | 19 | 1B | 1D | 1F |
| 03 | 00 | 03 | 06 | 09 | 0C | 0F | 12 | 15 | 18 | 1B | 1E | 01 | 04 | 07 | 0A | 0D | 10 | 13 | 16 | 19 | 1C | 1F | 02 | 05 | 08 | 0B | 0E | 11 | 14 | 17 | 1A | 1D | 20 |
| 04 | 00 | 04 | 08 | 0C | 10 | 14 | 18 | 1C | 20 | 01 | 05 | 09 | 0D | 11 | 15 | 19 | 1D | 02 | 06 | 0A | 0E | 12 | 16 | 1A | 1E | 03 | 07 | 0B | 0F | 13 | 17 | 1B | 1F |
| 05 | 00 | 05 | 0A | 0F | 14 | 19 | 1E | 01 | 06 | 0B | 10 | 15 | 1A | 1F | 02 | 07 | 0C | 11 | 16 | 1B | 20 | 03 | 08 | 0D | 12 | 17 | 1C | 04 | 09 | 0E | 13 | 18 | 1D |
| 06 | 00 | 06 | 0C | 12 | 18 | 1E | 01 | 07 | 0D | 13 | 19 | 1F | 02 | 08 | 0E | 14 | 1A | 20 | 03 | 09 | 0F | 15 | 1B | 04 | 0A | 10 | 16 | 1C | 05 | 0B | 11 | 17 | 1D |
| 07 | 00 | 07 | 0E | 15 | 1C | 01 | 08 | 0F | 16 | 1D | 02 | 09 | 10 | 17 | 1E | 03 | 0A | 11 | 18 | 1F | 04 | 0B | 12 | 19 | 20 | 05 | 0C | 13 | 1A | 06 | 0D | 14 | 1B |
| 08 | 00 | 08 | 10 | 18 | 20 | 01 | 09 | 11 | 19 | 02 | 0A | 12 | 1A | 03 | 0B | 13 | 1B | 04 | 0C | 14 | 1C | 05 | 0D | 15 | 1D | 06 | 0E | 16 | 1E | 07 | 0F | 17 | 1F |
| 09 | 00 | 09 | 12 | 1B | 01 | 0A | 13 | 1C | 02 | 0B | 14 | 1D | 03 | 0C | 15 | 1E | 04 | 0D | 16 | 1E | 05 | 0E | 17 | 20 | 06 | 0F | 18 | 07 | 10 | 19 | 08 | 11 | 1A |
| 0A | 00 | 0A | 14 | 1E | 01 | 0B | 15 | 1F | 02 | 0C | 16 | 20 | 03 | 0D | 17 | 04 | 0E | 18 | 05 | 0F | 19 | 06 | 10 | 1A | 07 | 11 | 1B | 08 | 12 | 1C | 09 | 13 | 1D |
| 0B | 00 | 0B | 16 | 01 | 0C | 17 | 02 | 0D | 18 | 03 | 0E | 19 | 04 | 0F | 1A | 05 | 10 | 1B | 06 | 11 | 1C | 07 | 12 | 1D | 08 | 13 | 1E | 09 | 14 | 1F | 0A | 15 | 20 |
| 0C | 00 | 0C | 18 | 01 | 0D | 19 | 02 | 0E | 1A | 03 | 0F | 1B | 04 | 10 | 1C | 05 | 11 | 1D | 06 | 12 | 1E | 07 | 13 | 1F | 08 | 14 | 20 | 09 | 15 | 0A | 16 | 0B | 17 |
| 0D | 00 | 0D | 1A | 01 | 0E | 1B | 02 | 0F | 1C | 03 | 10 | 1D | 04 | 11 | 1E | 05 | 12 | 1F | 06 | 13 | 20 | 07 | 14 | 08 | 15 | 09 | 16 | 0A | 17 | 0B | 18 | 0C | 19 |
| 0E | 00 | 0E | 1C | 01 | 0F | 1D | 02 | 10 | 1E | 03 | 11 | 1F | 04 | 12 | 20 | 05 | 13 | 06 | 14 | 07 | 15 | 08 | 16 | 09 | 17 | 0A | 18 | 0B | 19 | 0C | 1A | 0D | 1B |
| 0F | 00 | 0F | 1E | 01 | 10 | 1F | 02 | 11 | 20 | 03 | 12 | 04 | 13 | 05 | 14 | 06 | 15 | 07 | 16 | 08 | 17 | 09 | 18 | 0A | 19 | 0B | 1A | 0C | 1B | 0D | 1C | 0E | 1D |
| 10 | 00 | 10 | 20 | 01 | 11 | 02 | 12 | 03 | 13 | 04 | 14 | 05 | 15 | 06 | 16 | 07 | 17 | 08 | 18 | 09 | 19 | 0A | 1A | 0B | 1B | 0C | 1C | 0D | 1D | 0E | 1E | 0F | 1F |

注：PC-98XL/XA 内蔵型の固定ディスクインターフェイスを使用してフォーマットを行う場合，自動的に拡張フォーマット(トラックバッファ方式)となるため，インタリーブファクタは無視される．また PC-9801-27 を標準フォーマットでフォーマットする際には，インタリーブファクタは5でなければならない．

**(6) 使用上の注意**

① 同一ボリューム上に異なるインタリーブファクタのトラックを持つことはできない．コントローラは最初のアクセスでインタリーブファクタを記憶する．

② 出荷時の不良トラックは，不良トラックフラグを立て不良トラック扱いにしている．それゆえ，フォーマットを行うときにはこのトラックの先頭セクタを読み，不良トラックでないことを確認してからフォーマットしなければならない．

## 5.4.6　初期化(INITIALIZE)

**(1) 機能**

接続されている固定ディスク装置・インターフェイス，およびBIOS情報の初期設定を行う．

**(2) 入力**

・内部割り込みコード← 01BH

・AH ← BIOS コマンド識別コード(03H)

・AL ←デバイス種別(DA)(８×H)

ただし，×はチェックの対象にしない．どんな値でもよい．

**(3) 出力**

・CF ←0

・AH ←00H

**(4) 処理**

このコマンドにより，次の操作が行われる．

・ディスクコントローラ初期化．

・ユニットの接続状態をチェックし，READY 状態の装置に対応するシステム共通情報の Equipment Flag (DISK_EQUIP 55CH~55DH)を１(オン)にする．

・５インチ固定ディスクボード上のディップスイッチの状態を読み取り，その情報をコントローラに通知する(接続デバイスのディスク容量)．

・リトラクト(RETRACT)処理を行い，アームを未使用シリンダへ退避する．

## 5.4.7　ベリファイ(VERIFY)

**(1) 機能**

デバイス種別・ユニット番号(DA/UA)で指定されたデバイスの，指定されたディスクアドレスからデータを読み取り，読み取り動作ができることを確認する(メモリへ転送しない)．

(2) 入力

・内部割り込みコード← 01BH

・AL ← BIOS コマンド識別コード

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| × | × | r | × | 0 | 0 | 0 | 1 |

他は「5.4.1 データの読み取り(READ DATA)」と同じ.

(3) 出力

「5.4.1 データの読み取り(READ DATA)」と同じ.

(4) 処理

ディスク上の書き込んだデータが読み取れることを確認する.

## 5.4.8 センス(SENSE)

(1) 機能

指定されたデバイスの状態を通知する.

(2) 入力

・内部割り込みコード← 01BH

・AH ← BIOS コマンド識別コード04Hまたは84H

・AL ←デバイス種別・ユニット番号(DA/UA)(80H～81H)

(3) 出力

・CF ←終了条件

0：正常終了

1：異常終了

・AH ←ステータス情報(CF＝0の時に通知)

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| d7 | d6 | d5 | d4 | | | | |

| b3 | b2 | b1 | b0 | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 5MB | 接続されているデバイスの容量を通知する |
| 0 | 0 | 0 | 1 | 10MB | |
| 0 | 0 | 1 | 1 | 20MB | |
| 0 | 1 | 0 | 0 | 40MB | |

$d_7d_6d_5d_4$については「ステータス一覧」を参照.

・BX ←セクタ長(入力の AH ← 84H の時)

・CX ←シリンダ数(入力の AH ← 84H の時)

・DH ←ヘッド数(入力の AH ← 84H の時)

・DL ←セクタ数(入力の AH ← 84H の時)

**●サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;###########################################################################
;#                                                                         #
;#         ディスク装置（1MB TYPE）を初期化する                            #
;#                                                                         #
;###########################################################################
CODE     SEGMENT
         ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
         MOV     AX,SEG DATA              ;データセグメントを設定
         MOV     DS,AX
;                                         ;コントローラの初期化
         MOV     AH,3
         MOV     AL,90H                     ;UA/DA:1MBタイプの1台目
         INT     1BH
;
         JB      BAD
         MOV     DX,OFFSET OKMSG          ;CF=0:正常終了
         JMP     DISP
BAD:     MOV     DX,OFFSET BADMSG         ;CF=1:異常終了
;
DISP:    MOV     AH,9                     ;メッセージを表示
         INT     21H
EXIT:
         MOV     AH,4CH                   ;プログラムを終了
         INT     21H
CODE     ENDS

;#######  データ領域
DATA     SEGMENT
OKMSG    DB      0DH,0AH
         DB      '1MB タイプの1台目を初期化しました．',0DH,0AH,'$'
BADMSG   DB      0DH,0AH
         DB      '1MB タイプの1台目を初期化できませんでした．$'
DATA     ENDS

;#######  スタック領域
STACK    SEGMENT STACK
         DW      40H DUP (?)
STACK    ENDS

         END     START
```

```
;###########################################################################
;#                                                                         #
;#         ディスク装置（1MB TYPE）の状態を通知する                        #
;#                                                                         #
;###########################################################################
CODE     SEGMENT
         ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
         MOV     AX,SEG DATA              ;データセグメントを設定
         MOV     DS,AX
;                                         ;ディスク装置をセンス
         MOV     AH,84H
         MOV     AL,90H                     ;UA/DA:1MBタイプの1台目
         INT     1BH
         JB      BAD
;
         PUSH    AX                       ;NORMAL 時
         MOV     AH,9                       ;正常のメッセージを表示
         MOV     DX,OFFSET OKMSG
         INT     21H
         POP     AX

         TEST    AH,01H                     ;媒体は片面/両面 ?
         JNZ     RYOYOU
         MOV     DX,OFFSET OKST0            ;bit0=0:片面
         JMP     DISP
;
RYOYOU:  MOV     DX,OFFSET OKST1            ;bit0=1:両面
;
DISP:    MOV     AH,9                       ;片面/両面を表示
         INT     21H
         JMP     EXIT
;
BAD:     PUSH    AX                       ;ABNORMAL 時
         MOV     AH,9
         MOV     DX,OFFSET BADMSG           ;異常のメッセージを表示
         INT     21H
         POP     AX
;
NR:      AND     AH,0F0H
         CMP     AH,60H                     ;NOT READY ?
         JNZ     WP
         MOV     AH,9                       ;NOT READY を表示
         MOV     DX,OFFSET BADMSG1
         INT     21H
;
WP:      AND     AH,0F0H
         CMP     AH,70H                     ;NOT WRITABLE ?
         JNZ     EXIT
         MOV     AH,9                       ;WRITE PROTECTED を表示
         MOV     DX,OFFSET BADMSG2
         INT     21H
EXIT:
```

```
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;####### データ領域
DATA    SEGMENT
OKMSG   DB      ØDH,ØAH,'1MB タイプのドライブの1台目は$'
OKSTØ   DB      '1MB 専用タイプです.',ØDH,ØAH,'$'
OKST1   DB      '1MB / 64ØKB 両用タイプです.',ØDH,ØAH,'$'
BADMSG  DB      ØDH,ØAH,'ドライブをセンスできませんでした.$'
BADMSG1 DB      ØDH,ØAH,'媒体がセットされていません.',ØDH,ØAH,'$'
BADMSG2 DB      ØDH,ØAH,'WRITE PROTECT 状態です.',ØDH,ØAH,'$'
DATA    ENDS

;####### スタック領域
STACK   SEGMENT STACK
        DW      4ØH DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                                #
;#      指定するセクタからデータを読み込む                        #
;#                                                                #
;##################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;                                       ;データの読み込み
        MOV     BX,1Ø24                   ;データ長   :1Ø24 Byte
        MOV     CH,3                      ;セクタ長   :3(1Ø24)
        MOV     CL,Ø                      ;シリンダ番号:Ø
        MOV     DH,1                      ;ヘッド番号:1
        MOV     DL,4                      ;セクタ番号:4
        MOV     AX,SEG WORK               ;データ領域セグメント
        MOV     ES,AX
        MOV     BP,OFFSET WORK            ;データ領域オフセット
        MOV     AH,ØD6H                   ;
        MOV     AL,9ØH                    ;DA/UA: 1MB ,1台目
        INT     1BH
        JB      BAD
;                                       ;NORMAL 時
        MOV     AH,9                      ;読み込み完了を表示
        MOV     DX,OFFSET OKMSG
        INT     21H
        JMP     EXIT
;                                       ;ABNORMAL 時
BAD:    MOV     CL,4
        SHR     AH,CL                     ;ステータスを編集
        ADD     AH,3ØH
        MOV     STATUS,AH
        MOV     AH,9                      ;ステータスを表示
        MOV     DX,OFFSET BADMSG
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;####### データ領域
DATA    SEGMENT
OKMSG   DB      'データの読み込みが完了しました.',ØDH,ØAH,'$'
BADMSG  DB      'データの読み込みができませんでした.',ØDH,ØAH
        DB      'ステータスは '
STATUS  DB      ?,'$'
        EVEN
WORK    DB      1Ø24H DUP(?)
DATA    ENDS
;####### スタック領域
STACK   SEGMENT STACK
        DW      1ØØ DUP (?)
STACK   ENDS
        END     START
```

```
;###################################################################
;#                                                                 #
;#        指定するセクタにデータを書き出す                           #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;                                       ;セクタの書き出し
        MOV     BX,1Ø24                    ;データ長      :1Ø24
        MOV     CH,3                       ;セクタ長      :3(1Ø24)
        MOV     CL,Ø                       ;シリンダ番号:Ø
        MOV     DH,1                       ;ヘッド番号  :1
        MOV     DL,4                       ;セクタ番号  :4
        MOV     AX,SEG WORK                ;データ領域セグメント
        MOV     ES,AX
        MOV     BP,OFFSET WORK             ;データ領域オフセット
        MOV     AH,ØD5H
        MOV     AL,9ØH                     ;UA/DA:1MB, 1台目
        INT     1BH
        JB      BAD
;                                       ;NORMAL 時
        MOV     AH,9
        MOV     DX,OFFSET OKMSG            ;書き出し完了を表示
        INT     21H
        JMP     EXIT
;                                       ;ABNORMAL 時
BAD:    MOV     CL,4                       ;ステータスを編集
        SHR     AH,CL
        ADD     AH,3ØH
        MOV     STATUS,AH
        MOV     AH,9                       ;ステータスを表示
        MOV     DX,OFFSET BADMSG
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;####### データ領域
DATA    SEGMENT
OKMSG   DB      'DATA の書き込みが完了しました.',ØDH,ØAH,'$'
BADMSG  DB      'DATA の書き込みができませんでした.'
STATUS  DB      ?,ØDH,ØAH,'$'
WORK    DB      1Ø24H DUP(55H)
DATA    ENDS
;#######  スタック領域
STACK   SEGMENT STACK
        DW      1ØØH DUP (?)
STACK   ENDS
        END     START
```

```
;###################################################################
;#                                                                 #
;#        固定ディスク装置の初期化                                   #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;                                       ;コントローラの初期化
        MOV     AH,3
        MOV     AL,8ØH                     ;UA/DA:5"HD 1台目
        INT     1BH
        JB      BAD
;
        MOV     DX,OFFSET OKMSG         ;CF=Ø:正常終了
        JMP     DISP
BAD:    MOV     DX,OFFSET BADMSG        ;CF=1:異常終了
;
DISP:   MOV     AH,9                    ;メッセージを表示
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;####### データ領域
DATA    SEGMENT
OKMSG   DB      '固定ディスク装置を初期化しました.'
        DB      ØDH,ØAH,'$'
BADMSG  DB      '固定ディスク装置を初期化できませんでした.'
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#######  スタック領域
STACK   SEGMENT STACK
        DW      4ØH DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                                #
;#      固定ディスク装置のリトラクト                                #
;#                                                                #
;##################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;                                       ;リトラクト
        MOV     AH,ØFH
        MOV     AL,8ØH                   ;UA/DA: 5"HD, 1台目
        INT     1BH
        JB      BAD
;
        MOV     DX,OFFSET OKMSG         ;CF=Ø:正常終了
        JMP     DISP
BAD:    MOV     DX,OFFSET BADMSG        ;CF=1:異常終了
;
DISP:   MOV     AH,9                    ;メッセージを表示
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#######  データ領域
DATA    SEGMENT
OKMSG   DB      'リトラクトを完了しました.',ØDH,ØAH,'$'
BADMSG  DB      'リトラクトに失敗しました.',ØDH,ØAH,'$'
DATA    ENDS

;#######  スタック領域
STACK   SEGMENT STACK
        DW      4ØH DUP (?)
STACK   ENDS

        END     START
```

```
;###########################################################
;#                                                         #
;#      指定するセクタにデータを書き出す                   #
;#                                                         #
;###########################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
                                        ;セクタの書き出し
        MOV     BX,1024                 ;データ長       1024
        MOV     CH,3                    ;セクタ長       3(1024)
        MOV     CL,0                    ;シリンダ番号   0
        MOV     DH,1                    ;ヘッド番号     1
        MOV     DL,4                    ;セクタ番号     4
        MOV     AX,SEG WORK             ;データ領域セグメント
        MOV     ES,AX
        MOV     BP,OFFSET WORK          ;データ領域オフセット
[illegible]                             ;UA/DA:1HD, 1台目
[illegible]
        JB      BAD
                                        ;NORMAL 時
[illegible]                             ;完了を表示
        INT     21H
[illegible]
                                        ;ABNORMAL 時
                                        ;ステータスを記憶
        MOV     STATUS,AH
                                        ;ステータスを表示
[illegible]
        INT     21H
[illegible]
;####### データ領域
DATA    SEGMENT
OKMSG   DB      'DATA の書き込みが完了しました',0DH,0AH,'$'
BADMSG  DB      'DATA の書き込みができませんでした',0DH,0AH,'$'
STATUS  DB      ?
WORK    DB      1024 DUP (55H)
DATA    ENDS
;####### スタック領域
STACK   SEGMENT STACK
        DW      100H DUP (?)
STACK   ENDS
        END     START
```

```
;###########################################################
;#                                                         #
;#      固定ディスク装置のリトラクト                       #
;#                                                         #
;###########################################################
```

```
;###########################################################
;#                                                         #
;#      固定ディスク装置の初期化                           #
;#                                                         #
;###########################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
                                        ;コントローラの初期化
        MOV     AH,3
        MOV     AL,80H                  ;UA/DA:5"HD 1台目
        INT     1BH
        JB      BAD

        MOV     DX,OFFSET OKMSG         ;CF=0:正常終了
        JMP     DISP
BAD:    MOV     DX,OFFSET BADMSG        ;CF=1:異常終了

DISP:   MOV     AH,9                    ;メッセージを表示
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;####### データ領域
DATA    SEGMENT
OKMSG   DB      '固定ディスク装置を初期化しました'
        DB      0DH,0AH,'$'
BADMSG  DB      '固定ディスク装置を初期化できませんでした'
        DB      0DH,0AH,'$'
DATA    ENDS

;####### スタック領域
STACK   SEGMENT STACK
        DW      40H DUP (?)
STACK   ENDS

        END     START
```

# 第6章
# マウスBIOS

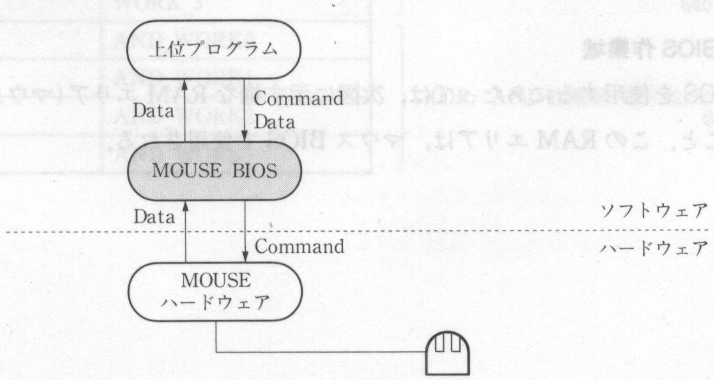
## ●マウス BIOS 機能一覧

| 機能コード | 機　　能 |
|---|---|
| 0 | 初期化 |
| 1 | カーソルの表示 |
| 2 | カーソルの消去 |
| 3 | カーソル位置の取得 |
| 4 | カーソル位置の設定 |
| 5 | ボタンの押下情報の取得 |
| 6 | ボタンの解放情報の取得 |
| 7 | 水平方向のカーソル移動範囲の設定 |
| 8 | 垂直方向のカーソル移動範囲の設定 |
| 9 | カーソルの形の設定 |
| 11 | マウスの移動距離の取得 |
| 12 | ユーザー定義サブルーチンのコール条件の設定 |
| 15 | ミッキー/ドット比の設定 |
| 16 | カーソル表示画面の設定 |
| 17 | マウス割り込みアドレスの設定 |
| 18 | マウス割り込みの許可 |

上記コードは10進数．また機能コード10，13，14は存在しない．

## 6.1 マウス BIOS 共通情報

### (1) マウス BIOS の位置付け



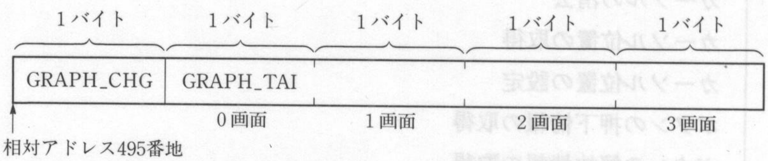
### (2) 上位プログラムとのインターフェイス

マウス BIOS は，上位プログラムからの内部割り込みによって起動される．

上位プログラムは「INT」を発行することによりマウス BIOS を呼出す(使用する INT ベクタに F800H(セグメント)，7FD0H(オフセット)を格納しておく．

マウス BIOS は，コマンド17(後述)によって，上位からもらった作業域のアドレスをシステム共通域の 5FC 番地に格納する．

マウス BIOS は，割り込みによって CG モードの書き替えを行うため，割り込みから抜ける時に，システム共通域の495番地の内容(GRAPH_CHG)の7ビット目が0であったら，GRAPH_CHG の内容でモード設定をし，7ビット目が1であったら，GRAPH_CHG の内容でモード設定をした後，システム共通域の496番地からの各画面のタイリング情報(GRAPH_TAI)を設定し直す．




GRAPH_CHG，GRAPH_TAI の初期値は00である．

マウス BIOS を使う時は，グラフ BIOS の初期化コマンド，もしくはテキスト画面の表示開始コマンドを先に実行すること．

### (3) システム共通域

| システム共通域<br>シンボル名・絶対番地 | 用　途 |
|---|---|
| MOUSEW<br>(5FCH) | マウス BIOS 作業域のセグメントアドレス |
| GRAPH_CHG<br>(495H) | グラフィック画面のモード情報(グラフィック画面の変更フラグ) |
| GRAPH_TAI<br>(496H～498H) | グラフィック画面のタイリング情報 |

### (4) マウス BIOS 作業域

マウス BIOS を使用するにあたっては，次図に示す様な RAM エリア(マウス BIOS 作業域)を確保すること．この RAM エリアは，マウス BIOS で使用される．

| ←1ワード→ | ←1ワード→ | |
|---|---|---|
| X_CURSOR | Y_CURSOR | ・カーソルの位置 |
| X_SW1_P | X_SW2_P | ・第1，第2ボタンの押された時の水平座標 |
| Y_SW1_P | Y_SW2_P | ・第1，第2ボタンの押された時の垂直座標 |
| X_SW1_R | X_SW2_R | ・第1，第2ボタンの離された時の水平座標 |
| Y_SW1_R | Y_SW2_R | ・第1，第2ボタンの離された時の垂直座標 |
| REL_X_MICKEY | REL_Y_MICKEY | ・水平方向，垂直方向マウス移動距離 |
| CUR_X_MIC | CUR_Y_MIC | ・ミッキー/ピクセル比 |
| X_CURSOR_CENTER | Y_CURSOR_CENTER | ・カーソルの中心座標 |
| CHANGED | UPPER_C | ・カーソル変更フラグ・マスク上部カウント |
| CURSOR_ADRS | CURSOR_ADRS_2 | ・カーソルアドレス |
| SCREEN_OFFSET | LO_C | ・スクリーンアドレス・マスク下部カウント |
| SWITCH_1 | SWITCH_2 | ・第1，第2ボタンの状態 |
| SW1_P_C | SW2_P_C | ・第1，第2ボタンの押された回数 |
| SW1_R_C | SW2_R_C | ・第1，第2ボタンの離された回数 |
| INT_COND | INT_MASK | ・ユーザー定義サブルーチンコール条件 |
| USER_PROC | | ・ユーザー定義サブルーチンのアドレス |
| NEW_CUR_OFF | CUR_OFF | ・カーソルオフセット |
| TIMER | GAMN_OLD | ・タイマ　・前スクリーン情報 |
| SEG1 | CUR_BLO_CO | ・カーソルブロックの大きさ |
| SEG2 | SEG3 | ・カーソルデータ転送回数 |
| INT_COUNT | INT_COUNT_C | ・8255からの割り込み間隔 |
| WAIT | SAVED_CURSOR_ON | ・V_SYNCフラグ　・カーソルセーブフラグ |
| CUR_CUR_ON | NEW_SCREEN_OFF | ・前のカーソルの状態　・新カーソルの位置 |
| CURSOR_ON | AND_XOR | ・カーソルの状態　・AND，XORマスクビット |
| GAMEN | CUR_SHIFT_C | ・カーソル表示情報　・カーソルシフトカウント |
| MIN_HORIZONTAL | MAX_HORIZONTAL | ・水平値(最小，最大) |
| MIN_VERTICAL | MAX_VERTICAL | ・垂直値(最小，最大) |
| VRAM_SEG | VRAM_OLD_SEG | ・カーソル表示セグメント・前カーソル表示セグメント |
| CURSOR_HEIGHT | VERTICAL_LINES | ・カーソルの高さ　・½水平値 |
| SCREEN_OLD_OFFSET | CUR_OLD_HEIGHT | ・前スクリーンのアドレス・前カーソルの高さ |
| X_MICKEY | Y_MICKEY | ・ミッキー値 |
| WORK_0 | WORK_1 | 各画面セーブ領域(40×32)ビット×4　640バイト |
| WORK_2 | WORK_3 | |
| XOR_WORK0 | AND_WORK0 | XOR，AND作業域(40×32)ビット×4　640バイト |
| XOR_WORK1 | AND_WORK1 | |
| XOR_WORK2 | AND_WORK2 | |
| XOR_WORK3 | AND_WORK3 | |

## 6.2　初期化

(1) 入力

・AX ← 0 (機能コード)

(2) 出力

・AX ← 環境状態

　0：マウスを使用できない環境

　−1：マウスを使用できる環境

・BX ← ボタン数(2)

AX，BX 以外のすべてのレジスタが保証される．

(3) 説明

カーソルの表示，カーソルの形，カーソルの中心点，ミッキー／ドット比，マウスの割り込み周期等の環境を初期化する．

マウス環境の初期設定は次のとおり．

・カーソル表示：表示しない

・カーソル位置：スクリーンの中心の位置(512，384)

・カーソルの形：左上向きの矢印

・カーソルの中心点：(0，0)

・カーソル移動範囲：スクリーン全体

　水平方向：0～1119

　垂直方向：0～935

・ミッキー／ドット比：水平方向，垂直方向ともに8

・マウスの割り込み周期：8.33ms(120Hz)

## 6.3　カーソルの表示

(1) 入力

・AX ← 1 (機能コード)

(2) 出力

なし

AX 以外のすべてのレジスタが保証される．

(3) 説明

BIOS作業域のCURSOR_ON，CUR_CUR_ON，WAIT，CHANGEDフィールドをON(FFFFH)にし，カーソルをスクリーン上に表示させる．1度この機能を実行すると，カーソルの消去を実行するまで，カーソルがマウスの動きに従ってスクリーン上を動く．

## 6.4 カーソルの消去

(1) 入力

・AX ← 2(機能コード)

(2) 出力

なし

AX以外のすべてのレジスタが保証される．

(3) 説明

BIOS作業域のCURSOR_ON, CUR_CUR_ONフィールドをOFF(0000H), CHANGEDフィールドをON(FFFFH)にし，スクリーン上に表示されているカーソルを消す．カーソルは，1度消されるとカーソル表示を実行するまで，スクリーンには表示されない．しかし，カーソルは表示されていない間も，マウスを動かすことによってカーソルはスクリーン上で常に移動している．

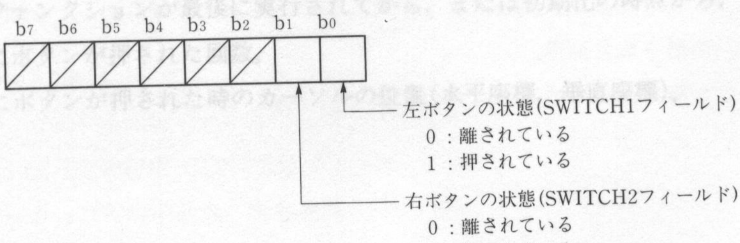
## 6.5 カーソル位置の取得

(1) 入力

・AX ← 3(機能コード)

(2) 出力

・BX ← ボタンの状態



・CX ← カーソルの位置の水平座標(0～1119)

(X_CURSOR フィールド)

・DX ← カーソルの位置の垂直座標(0～935)

(Y_CURSOR フィールド)

BX，CX，DX 以外のレジスタは保証される．

(3)説明

現在のカーソルの位置を得る．カーソルの位置は，水平座標，垂直座標で得られ，それはカーソルの移動範囲内の値である．また，この時のマウスの右ボタン，および左ボタンの状態(押されているか，押されていない)も得ることができる．

## 6.6　カーソル位置の設定

(1) 入力

・AX ← 4 (機能コード)

・CX ← カーソルの新しい位置の水平座標(0～1119)

・DX ← カーソルの新しい位置の垂直座標(0～935)

(2) 出力

なし

(3) 説明

カーソルの位置を指定の位置に設定する．アプリケーションプログラムが，希望の水平座標，垂直座標を指定して，このファンクションを実行すると，X_CURSOR，Y_CURSOR フィールドに値がセットされ，カーソルはその位置に移動する．希望の位置がカーソル移動範囲外の場合には，移動範囲内の端にカーソルが移動する．

CHANGED フィールドは ON(FFFFH)になる．

## 6.7 ボタンの押下情報の取得

**(1) 入力**

・AX ← 5(機能コード)

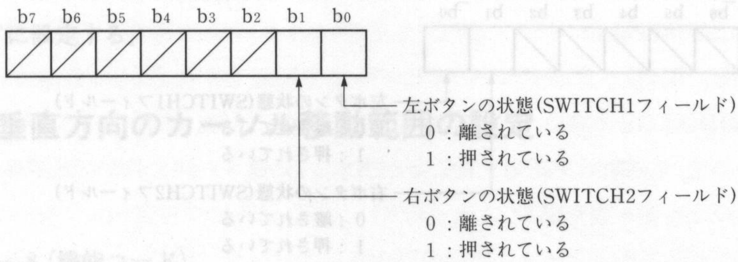
・BX ← ボタンの指定

0：左ボタン

1：右ボタン

**(2) 出力**

・AX ← ボタンの状態




・BX ← ボタンが押された回数(SW1_P_CまたはSW2_P_Cフィールド)

・CX ← 最後にボタンが押された時の，カーソル位置の水平座標

(X_SW1_PまたはX_SW2_Pフィールド)

・DX ← 最後にボタンが押された時の，カーソル位置の垂直座標

(Y_SW1_PまたはY_SW2_Pフィールド)

**(3) 説明**

マウスのボタンの押下(押されること)に関する各種の情報を取得する．取得できる情報は，次のものである．

・ボタンの現在の状態．

・このファンクションが最後に実行されてから，または初期化の時点から，今回実行されるまでにボタンが押された回数．

・最後にボタンが押された時のカーソルの位置(水平座標，垂直座標)．

## 6.8 ボタンの解放情報の取得

**(1) 入力**

・AX ← 6 (機能コード)

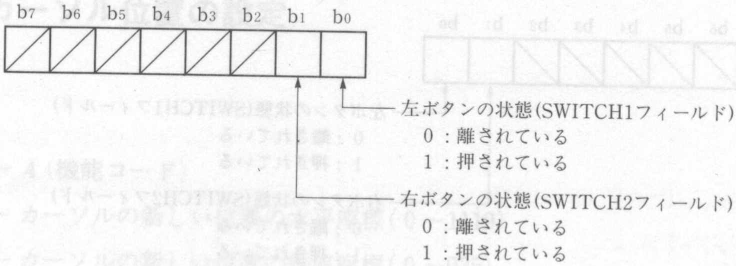
・BX ← ボタンの指定

　　0 ：左ボタン

　　1 ：右ボタン

**(2) 出力**

・AX ←ボタンの状態




・BX ← ボタンが離された回数(SW1_R_CまたはSW2_R_Cフィールド)

・CX ← 最後にボタンが離された時の，カーソル位置の水平座標

　(X_SW1_RまたはX_SW2_Rフィールド)

・DX ← 最後にボタンが離された時の，カーソル位置の垂直座標

　(Y_SW1_RまたはY_SW2_Rフィールド)

**(3) 説明**

マウスのボタンの解放(離されること)に関する各種の情報を取得する．次の情報が取得できる．

・ボタンの現在の位置．

・このファンクションが最後に実行されてから，または初期化の時点から，今回実行されるまでにボタンが離された回数．

・最後にボタンが離された時のカーソルの位置(水平座標，垂直座標)

## 6.9　水平方向のカーソル移動範囲の設定

(1) 入力

・AX ← 7 (機能コード)

(2) 出力

・CX ← カーソルの水平方向の移動範囲の最小値

・DX ← カーソルの水平方向の移動範囲の最大値

AX 以外のすべてのレジスタが保証される.

(3) 説明

カーソルの水平方向の移動範囲( 0 ～1119) を MIN_HORIZONTAL, MAX_HORIZONTAL フィールドに設定する.

## 6.10　垂直方向のカーソル移動範囲の設定

(1) 入力

・AX ← 8 (機能コード)

(2) 出力

・CX ←カーソルの垂直方向の移動範囲の最小値

・DX ←カーソルの垂直方向の移動範囲の最大値

AX 以外のすべてのレジスタが保証される.

(3) 説明

カーソルの垂直方向の移動範囲( 0 ～935) を MIN_VERTICAL, MAX_VERTICAL フィールドに設定する.

## 6.11 カーソルの形の設定

(1) 機能

・AX ← 9 (機能コード)

・BX ← カーソルの中心点の水平座標(－16～32)

(X_CURSOR_CENTER フィールド)

・CX ← カーソルの中心点の垂直座標(－16～48)

(Y_CURSOR_CENTER フィールド)

・ES/DX ← カーソルの形を決定するデータのアドレス
　　データの形式は32×32ビット
　　(S_REP，Y_REP フィールド)
・ES/SI ← カーソルブロックの値
　　(CURSOR_ADRS_2 フィールド)

**(2) 出力**

AX 以外のすべてのレジスタが保証される．

**(3) 説明**

カーソルの形や中心点を設定する．カーソルの形は，四角形のカーソルのうちの，どのドットを表示するかによって決まる．つまり表示されたドットの集合がカーソルとして見える．カーソルの中心点の座標指定は，カーソルの左上点を(0，0)とした座標系の位置で与える．

カーソル表示で，AND，XORのマスクパターンをとることが出来る．両方のパターンを使用する場合，AND，XORの順にカーソルの形を設定する．また，各画面(PC-98XL/XAは4プレーンを持つので，4画面分を指定することが出来る)に違ったパターンを表示させる場合には4画面分のカーソル形を設定する．

**例1**　4画面に違ったカーソルパターンで，AND，XORをとる場合．

ES : DX →

| | |
|---|---|
| 0画面の<br>XOR用カーソル形 | 0画面の<br>AND用カーソル形 |
| 1画面の<br>XOR用カーソル形 | 1画面の<br>AND用カーソル形 |
| 2画面の<br>XOR用カーソル形 | 2画面の<br>AND用カーソル形 |
| 3画面の<br>XOR用カーソル形 | 3画面の<br>AND用カーソル形 |

**例2**　4画面に違ったカーソルパターンで，XORをとる場合．

ES : DX →

| | |
|---|---|
| 0画面の<br>XOR用カーソル形 | 1画面の<br>XOR用カーソル形 |
| 2画面の<br>XOR用カーソル形 | 3画面の<br>XOR用カーソル形 |

例3　1画面にAND，XORをとる場合．

ES : DX →

| XOR用カーソル形 | AND用カーソル形 |
|---|---|

例4　1画面にXORをとる場合．

ES : DX →

| XOR用カーソル形 |
|---|

1画面にカーソルを描くか，4画面にカーソルを描くか，またXORだけかANDとXORをとって描くかの指定はコマンド16を使用する．

ES/SIには上位にカーソルブロックの横のビット数，下位に縦のビット数を入れる．なお，最小パターンは8×8ビットで，縦横ともバイト単位で大きくすることができ，最大32×32ビットである（マウスBIOS作業域の各画面セーブ領域は，どんな時でも(32×32)×4，AND，XOR作業域は(40×32ビット)×カーソルパターン数だけ必要である）．

カーソルパターンの変更時（AND，XORパターンからXORパターンへ，XORパターンからAND，XORパターンへ）は，カーソルの形の設定の前にカーソル表示画面の設定（6.15参照）を実行する必要がある．

注：ES:DXで指定されたカーソルパターンはBIOSで参照するので，次の新しいカーソルパターンに変更するまでは内容を保証すること．

## 6.12　マウスの移動距離の取得

**(1) 入力**

・AX　←11（機能コード）

**(2) 出力**

・CX　←マウスの水平方向の移動距離（−1119～1119）
　　（REL_X_MICKEYフィールド）

・DX　←マウスの垂直方向の移動距離（−935～935）
　　（REL_Y_MICKEYフィールド）

AX，CX，DX以外のレジスタは保証される．

**(3) 説明**

マウスの移動距離を取得する．このファンクションが最後に実行された時のマウスの位置から，今回実行した時点でのマウスの位置までの，水平方向および垂直方向の相対的な距離が通知される．水平方向では右の向きが正であり，垂直方向では，手前の向きが正となる．距離の単位はミッキーである．

## 6.13　ユーザー定義サブルーチンのコール条件の設定

**(1) 入力**

・AX ← 12(機能コード)
・CX ← コール条件
　　ビット0：カーソル位置の変化
　　ビット1：左ボタンが押される
　　ビット2：左ボタンが離される
　　ビット3：右ボタンが押される
　　ビット4：右ボタンが離される
　　ビット5～15：未使用

(最下位ビットをビット0として，各ビットが1の時はコールし，0の時はコールしない)

・ES/DX ← ユーザー定義サブルーチンのアドレス

**(2) 出力**

なし

AX以外のすべてのレジスタが保証される．

**(3) 説明**

ユーザーが定義したサブルーチンをコールする条件(INT_MASKフィールド)と，そのサブルーチンのアドレス(USER_PROCフィールド)を設定する．マウスBIOSは，次の5つの現象が発生した時に，サブルーチンをコールすることができる．

・カーソルの位置が変わった場合
・左ボタンが押された場合
・左ボタンが離された場合
・右ボタンが押された場合
・右ボタンが離された場合

これらの現象のうち，どれが発生した時にサブルーチンをコールするかをこのファンクションによって設定する．コール条件は5つの現象のうち重複して指定でき，そのうちの1つでも発生すれば，サブルーチンがコールされる．

マウスBIOSは次の手順でサブルーチンをコールする。

①マウスから割り込みが発生すると，制御がマウスBIOSに移る。
②マウスBIOSは，コール条件が満たされているかどうかを調べる。
③満足されていない場合には，そのまま次の動作に移るが，満足されている場合にはCALL FAR_PROC命令によってサブルーチンに制御を移す。

したがってサブルーチンからマウスBIOSに制御を戻すためには，RET FAR_PROC命令を使用しなければならない。

また，サブルーチンをコールする時には，各レジスタには次の情報が格納されている。

・AX ← コールの原因となった現象
　　1：カーソルの位置が変わった
　　2：左ボタンが押された
　　4：左ボタンが離された
　　8：右ボタンが押された
　　16：右ボタンが離された
・BL ← 左ボタンの状態
　　0：離されている
　　−1：押されている
・BH ← 右ボタンの状態
　　0：離されている
　　−1：押されている
・CX ← カーソルの位置の水平座標
・DX ← カーソルの位置の垂直座標

## 6.14　ミッキー／ドット比の設定

**(1) 入力**

・AX ← 15(機能コード)
・CX ← 水平方向のミッキー／ドット比
・DX ← 垂直方向のミッキー／ドット比

**(2) 出力**

なし

AX以外のすべてのレジスタが保証される。

**(3) 説明**

マウスの移動距離と，それに対応するカーソルの移動距離との比を設定する．水平方向及び垂直方向に，カーソルが8ドット移動するのに要するマウスの水平方向及び垂直方向の移動距離(ミッキー／ドット比)を単位としてX_MICKEY, Y_MICKEYフィールドに設定する(ミッキー値)．これにより，マウスを少し動かしただけでカーソルが大きく移動したり，逆にマウスをかなり動かしてもカーソルは少ししか移動しないというように，マウスの感度を変えることができる．ミッキー／ドット比の指定範囲は1～32767．0の場合は前回の値を保証する．

## 6.15 カーソル表示画面の設定

**(1) 入力**

・AX ← 16(機能コード)

・BX ← カーソルの描画状態(詳しくは(4)を参照)

**(2) 出力**

なし

AX以外のすべてのレジスタが保証される．

**(3) 説明**

カーソルの表示画面，表示パターンを設定する．

**(4) カーソル描画状態**(GAMENフィールド)

ビット0：プレーン0への描画
ビット1：プレーン1への描画
ビット2：プレーン2への描画
ビット3：プレーン3への描画
ビット4：カーソル形4個
ビット5：カーソル形1個
ビット6：未使用
ビット7：AND, XORマスク
ビット8～ビット15：未使用

**例1**　1個のカーソルを1つの画面にXORだけとって描く場合

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BX | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

**例2**　1個のカーソルを4つの画面にAND, XORをとって描く場合

| | $b_{15}$ | $b_{14}$ | $b_{13}$ | $b_{12}$ | $b_{11}$ | $b_{10}$ | $b_9$ | $b_8$ | $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BX | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

**例3**　4個のカーソルを4つの画面にAND, XORをとって描く場合

| | $b_{15}$ | $b_{14}$ | $b_{13}$ | $b_{12}$ | $b_{11}$ | $b_{10}$ | $b_9$ | $b_8$ | $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BX | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

注：1個のカーソル形を2つの画面に描くことは出来ない.

4個のカーソル形は4つの画面に描かなければならない.

この時のカーソル形のアドレスの格納方法に付いては「6.11 カーソルの形の設定」を参照すること.

## 6.16　マウス割り込みアドレスの設定

### (1) 入力

・AX ← 17(機能コード)

・BX ← マウスBIOS作業域のセグメントベース

・CX ← 割り込み間隔(1～FFH)

### (2) 出力

なし

AX以外のすべてのレジスタが保証される.

### (3) 説明

マウスを使用する際には, 最初にこのファンクションを実行しなければならない.

割り込みベクタテーブルにマウス割り込み処理ルーチン(INT2のハードウェア割り込み)のアドレスを設定し, 8259の割り込み許可ビットのマスクを解除する.

マウスBIOS作業域のセグメントベースをMOUSEWフィールドに設定し, 何回に1回インターフェイスボードからの割り込みを受け付けるか指示する値(割り込み間隔)をINT_COUNTフィールドに設定する. 標準の8.33msec間隔に設定する場合には1を指定する.

## 6.17　マウス割り込みの許可

### (1) 入力

・AX ← 18(機能コード)

・BX ← 8255割り込み設定

　　1 : 8255からの割り込み停止

　　0 : 8255からの割り込み開始

### (2) 出力

なし

AX以外のすべてのレジスタは保証される.

### (3) 説明

8255からの割り込みを停止／開始する.

コマンド0を行った時点で割り込みは開始されている.

● **サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;##################################################################
;#                                                                #
;#   Mouse BIOSを初期化（割込みベクタ 4ØH を使用）                #
;#                                                                #
;##################################################################
MOUSE     GROUP    CODE,INTRPT,GRAPHW,MOUSEW
CODE      SEGMENT  PARA PUBLIC 'MOUSE'
          ASSUME   CS:CODE,SS:STACK,ES:MOUSEW
START:
          MOV      AX,SEG GRAPHW               ;作業領域セグメントを設定
          MOV      DS,AX
;                                              ;GRAPH BIOS の初期設定
GINIT:    MOV      WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
          MOV      WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
          MOV      AH,Ø
          INT      1DH
;                                              ;グラフ画面のモードを設定
GSCREEN:MOV        BYTE PTR DS:[1ØØH],Ø        ;表示モード:カラー
          MOV      BYTE PTR DS:[1Ø1H],Ø        ;表示SW:表示あり,普通
          MOV      BYTE PTR DS:[1Ø2H],Ø        ;アクティブページ  :Ø
          MOV      BYTE PTR DS:[1Ø3H],1        ;ディスプレイページ:1
          MOV      AH,1
          INT      1DH
;                                              ;描画領域を設定し,塗る
GVIEW:    MOV      WORD PTR DS:[1ØØH],Ø        ;描画領域:
          MOV      WORD PTR DS:[1Ø2H],Ø        ;  (Ø,Ø)-(1119,749)
          MOV      WORD PTR DS:[1Ø4H],1119
          MOV      WORD PTR DS:[1Ø6H],749
          MOV      BYTE PTR DS:[1Ø8H],ØØH      ;描画領域色 :ØØH
          MOV      BYTE PTR DS:[1Ø9H],ØFFH     ;外枠は描かない
          MOV      AH,2
          INT      1DH
;                                              ;割込みベクタをセット
          MOV      AX,SEG MOUSEW
          MOV      ES,AX
          PUSH     DS
          MOV      AX,ØF8ØØH
          MOV      DS,AX                       ;割込みセグメント
          MOV      DX,7FDØH                    ;割込みオフセット
          MOV      AH,25H
          MOV      AL,4ØH                      ;割込みタイプ:4ØH
          INT      21H
          POP      DS
;                                              ;MOUSE 割込み間隔設定
          MOV      AX,SEG MOUSEW
          MOV      BX,AX                       ;MOUSE BIOS 作業領域
          MOV      AX,17
          MOV      CX,1                        ;割込み間隔 : 1
          INT      4ØH
;                                              ;MOUSE BIOS 作業域初期設定
          MOV      AX,Ø
          INT      4ØH
;                                              ;マウス割込みを STOP
          MOV      AX,18
          MOV      BX,1
          INT      4ØH
EXIT:
          MOV      AH,4CH                      ;プログラムを終了
          INT      21H
CODE      ENDS

;######### 中断処理ルーチン
INTRPT    SEGMENT  PARA PUBLIC 'MOUSE'
          ASSUME   CS:INTRPT,DS:INTRPT
INTR      PROC     FAR
          RET
INTR      ENDP
INTRPT    ENDS

;######### GRAPH BIOS  使用作業領域
GRAPHW    SEGMENT  PARA PUBLIC 'MOUSE'
GBIOS     DB       256 DUP(?)                  ;GRAPH BIOS UCW
PARALST   DB       128 DUP(?)                  ;パラメータリスト
GWORK     DB       512 DUP(?)                  ;GRAPH BIOS 作業領域
GRAPHW    ENDS

;######### MOUSE BIOS 使用作業領域
MOUSEW    SEGMENT  PARA PUBLIC 'MOUSE'
MW        DB        2ØØØ DUP(Ø)
MOUSEW    ENDS

;######### スタック領域
STACK     SEGMENT  PARA STACK 'STACK'
          DW       1ØØH DUP (?)
STACK     ENDS

          END      START
```

```
;##################################################
;#                                                #
;#      マウスカーソルを表示する                  #
;#                                                #
;##################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW           ;作業領域セグメントを設定
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0       ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0       ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0       ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1       ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0       ;描画領域:
        MOV     WORD PTR DS:[102H],0       ;  (00)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H     ;描画領域色 :00H
        MOV     BYTE PTR DS:[109H],0FFH    ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                      ;割込みセグメント
        MOV     DX,7FD0H                   ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                     ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS           ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                       ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                   ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                       ;割込み間隔 : 1
        INT     40H
;                                       ;MOUSE BIOS 作業域初期設定
        MOV     AX,0
        INT     40H
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     40H
;                                       ;データセグメント設定
        MOV     AX,SEG DATA
        MOV     DS,AX
;                                       ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,08H                  ;キー入力待ち
        INT     21H
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     40H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG     DB      '何かキーを押すと,マウス割込みが停止します.'
        DB      0dh,0ah,'$'
DATA    ENDS
;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
;#########  GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS
;#########  MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2000 DUP(0)
MOUSEW  ENDS
;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS
        END     START
```

```
;###############################################################
;#                                                             #
;#      マウスカーソルの表示を停止する                           #
;#                                                             #
;###############################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW           ;作業領域セグメントを設定
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0           ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0           ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0           ;アクティブページ   :0
        MOV     BYTE PTR DS:[103H],1           ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0           ;描画領域:
        MOV     WORD PTR DS:[102H],0           ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H         ;描画領域色  :00H
        MOV     BYTE PTR DS:[109H],0FFH        ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                           ;割込みセグメント
        MOV     DX,7FD0H                        ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                          ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS           ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                       ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                           ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                            ;割込み間隔 : 1
        INT     40H
;                                       ;MOUSE BIOS 作業域初期設定
        MOV     AX,0
        INT     40H
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     40H
;                                       ;データセグメント設定
        MOV     AX,SEG DATA
        MOV     DS,AX
;                                       ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,08H                  ;キー入力待ち
        INT     21H
;                                       ;マウスカーソルの表示停止
        MOV     AX,2
        INT     40H
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     40H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS
;#########      データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG     DB      '何かキーを押すと,マウスカーソルが消えます.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########      中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########      GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########      MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB        2000 DUP(0)
MOUSEW  ENDS
```

```
;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;##########################################################################
;#                                                                        #
;#          マウスカーソルの位置を通知する                                  #
;#                                                                        #
;##########################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX
;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR  ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR     ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H          ;描画領域色  :00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                           ;割込みセグメント
        MOV     DX,7FD0H                        ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                          ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLEAR                 ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                           ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                            ;割込み間隔 : 1
        INT     40H
```

```
;                                       ;MOUSE BIO作業域初期設定
          MOV     AX,0
          INT     40H
;                                       ;MICKEY/PIXEL 比を設定
          MOV     AX,15
          MOV     CX,10                   ;水平方向 :10
          MOV     DX,10                   ;垂直方向 :10
          INT     40H
;                                       ;マウスカーソルを表示
          MOV     AX,1
          INT     40H
;                                       ;カーソル位置を取得
POS:      MOV     AX,3
          INT     40H
          MOV     STOREX,CX               ;水平座標
          MOV     STOREY,DX               ;垂直座標
;                                       ;水平座標を編集
          MOV     BX,STOREX
          MOV     SI,OFFSET X
          CALL    CNVNUM
;                                       ;垂直座標を編集
          MOV     BX,STOREY
          MOV     SI,OFFSET Y
          CALL    CNVNUM
;                                       ;座標を表示
          MOV     AH,9
          MOV     DX,OFFSET MSG
          INT     21H
;                                       ;キー入力待ち
          MOV     AH,0BH
          INT     21H
          CMP     AL,0
          JE      POS
;                                       ;カーソルの表示を停止
EXIT:     MOV     AX,2
          INT     40H
;                                       ;マウス割込みを STOP
          MOV     AX,18
          MOV     BX,1
          INT     40H
;
          MOV     AH,9
          MOV     DX,OFFSET EC
          INT     21H
;                                       ;プログラムを終了
          MOV     AH,4CH
          INT     21H
;
;#########  座標の編集
CNVNUM:   MOV     CH,4                  ;BINARY ---> NUMERIC
          MOV     AX,BX
          ADD     SI,4
CVLOOP:   MOV     BL,10
          DIV     BL
          ADD     AH,30H
          MOV     DS:[SI],AH
          AND     AH,00H
          DEC     SI
          DEC     CH
          JNZ     CVLOOP
          RET

CODE      ENDS

;#########  データ領域
DATA      SEGMENT PARA PUBLIC 'MOUSE'
CLEAR     DB      1BH,'[2J',1BH,'[5>h','$'
MSG       DB      1BH,'[0;0H','現在のマウス・カーソルの位置は [ '
X         DB      5 DUP(' ')
          DB      ','
Y         DB      5 DUP(' ')
          DB      ' ] です.'
          DB      0DH,0AH,0DH,0AH
          DB      '何かキーを押すと終了します.',0DH,0AH,'$'
STOREX    DW      ?
STOREY    DW      ?
EC        DB      1BH,'[>5l','$'
DATA      ENDS

;#########  中断処理ルーチン
INTRPT    SEGMENT PARA PUBLIC 'MOUSE'
          ASSUME  CS:INTRPT,DS:INTRPT
INTR      PROC    FAR
          RET
INTR      ENDP
INTRPT    ENDS

;#########  GRAPH BIO 使用作業領域
GRAPHW    SEGMENT PARA PUBLIC 'MOUSE'
GBIOS     DB      256 DUP(?)            ;GRAPH BIO UCW
PARALST   DB      128 DUP(?)            ;パラメータリスト
GWORK     DB      512 DUP(?)            ;GRAPH BIO 作業領域
GRAPHW    ENDS

;#########  MOUSE BIO 使用作業領域
MOUSEW    SEGMENT PARA PUBLIC 'MOUSE'
MW        DB      2000 DUP(0)
MOUSEW    ENDS

;#########  スタック領域
STACK     SEGMENT PARA STACK 'STACK'
          DW      100H DUP (?)
STACK     ENDS

          END     START
```

```
;###############################################################
;#                                                             #
;#      マウスカーソルを移動する                               #
;#                                                             #
;###############################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                 ;作業領域セグメントを設定
        MOV     DS,AX
;                                             ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                             ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0           ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0           ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0           ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1           ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                             ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0           ;描画領域:
        MOV     WORD PTR DS:[102H],0           ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H         ;描画領域色  :00H
        MOV     BYTE PTR DS:[109H],0FFH        ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                             ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                          ;割込みセグメント
        MOV     DX,7FD0H                       ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                         ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                    ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS                  ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                             ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                          ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                           ;割込み間隔 : 1
        INT     40H
;                                             ;MOUSE BIOS 作業域初期設定
        MOV     AX,0
        INT     40H
;                                             ;マウスカーソルを表示
        MOV     AX,1
        INT     40H
;                                             ;メッセージを表示
        MOV     DX,OFFSET MSG1
        MOV     AH,09H
        INT     21H
        MOV     AH,08H                         ;キー入力待ち
        INT     21H
;                                             ;マウスカーソルを移動
        MOV     AX,4
        MOV     CX,0                           ;(0,0)
        MOV     DX,0
        INT     40H
;                                             ;メッセージを表示
        MOV     DX,OFFSET MSG2
        MOV     AH,09H
        INT     21H
        MOV     AH,01H                         ;キー入力待ち
        INT     21H
;                                             ;マウスカーソルの表示停止
        MOV     AX,2
        INT     40H
;                                             ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     40H
EXIT:
        MOV     AH,4CH                         ;プログラムを終了
        INT     21H
CODE    ENDS

;########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG1    DB      '何かキーを押すと,座標 [ 0,0 ] へ'
        DB      'マウスカーソルが移動します.',0DH,0AH,'$'
MSG2    DB      0DH,0AH
        DB      '何かキーを押すと,マウスカーソルが消えます.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
```

```
;######### GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;######### MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2000 DUP(0)
MOUSEW  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;########################################################################
;#                                                                      #
;#       マウスボタンの状態を通知する                                   #
;#                                                                      #
;########################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX
;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR  ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR     ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H          ;描画領域色  :00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                           ;割込みセグメント
        MOV     DX,7FD0H                        ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                          ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLEAR                 ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                           ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                            ;割込み間隔 : 1
        INT     40H
```

```
;                                    ;MOUSE BIOS 作業域初期設定
        MOV     AX,Ø
        INT     4ØH
;                                    ;マウスカーソルを表示
        MOV     AX,1
        INT     4ØH
;                                    ;マウスボタン押下を調査
POS:    MOV     AX,5
        MOV     BX,Ø                 ;左ボタンをオーダー
        INT     4ØH
        TEST    AX,2                 ;右ボタンDOWN:終了
        JNE     EXIT
        MOV     STOREX,CX            ;水平座標
        MOV     STOREY,DX            ;垂直座標
;                                    ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET X
        CALL    CNVNUM
;                                    ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET Y
        CALL    CNVNUM
;                                    ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        JMP     POS
EXIT:                                ;マウスカーソルを表示停止
        MOV     AX,2
        INT     4ØH
;                                    ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     4ØH
;
        MOV     DX,OFFSET EC
        MOV     AH,Ø9H
        INT     21H
;                                    ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;#########  座標の編集
CNVNUM: CMP     BX,1119
        JG      CNVRET
        CMP     BX,Ø
        JS      CNVRET
        ADD     SI,4
        MOV     CH,4                 ;BINARY ---> NUMERIC
        MOV     AX,BX
        MOV     BL,1Ø
CNVLOOP:DIV     BL
        ADD     AH,3ØH
        MOV     DS:[SI],AH
        AND     AH,ØØH
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLEAR   DB      1BH,'[2J',1BH,'[5>h','$'
MSG     DB      1BH,'[Ø;ØH','左ボタンを押した位置は［'
X       DB      5 DUP(' ')
        DB      ','
Y       DB      5 DUP(' ')
        DB      '］です.'
        DB      ØDH,ØAH,ØDH,ØAH
        DB      '右ボタンを押すと終了します','$'
STOREX  DW      ?
STOREY  DW      ?
EC      DB      1BH,'[>5l','$'

DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)           ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)           ;パラメータリスト
GWORK   DB      512 DUP(?)           ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########  MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2ØØØ DUP(Ø)
MOUSEW  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      1ØØH DUP (?)
STACK   ENDS

        END     START
```

```
;#################################################################
;#                                                               #
;#    マウスボタンの状態を通知する(ボタンを離した位置)            #
;#                                                               #
;#################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:  MOV     AX,SEG GRAPHW           ;作業領域セグメントを設定
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                       ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[1ØØH],Ø    ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø    ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø    ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1    ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[1ØØH],Ø    ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø    ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH  ;描画領域色  :ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;割込みベクタをセット
        MOV     AX,ØF8ØØH
        MOV     DS,AX                   ;割込みセグメント
        MOV     DX,7FDØH                ;割込みオフセット
        MOV     AH,25H
        MOV     AL,4ØH                  ;割込みタイプ:4ØH
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLEAR         ;画面をクリア
        MOV     AH,Ø9H
        INT     21H
;                                       ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                   ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                    ;割込み間隔 : 1
        INT     4ØH
;                                       ;MOUSE BIOS 作業域初期設定
        MOV     AX,Ø
        INT     4ØH
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     4ØH
;                                       ;マウスボタン解放を調査
POS:    MOV     AX,6
        MOV     BX,Ø                    ;左ボタンをオーダー
        INT     4ØH
        CMP     BX,Ø                    ;解放回数をチェック
        JE      KEYCHK
        MOV     STOREX,CX               ;水平座標
        MOV     STOREY,DX               ;垂直座標
;                                       ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET X
        CALL    CNVNUM
;                                       ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET Y
        CALL    CNVNUM
;                                       ;メッセージを表示
KEYCHK: MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
;                                       ;キー入力のチェック
        MOV     AH,ØBH
        INT     21H
        CMP     AL,Ø
        JE      POS
EXIT:                                   ;マウスカーソルを表示停止
        MOV     AX,2
        INT     4ØH
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     4ØH
;
        MOV     DX,OFFSET EC
        MOV     AH,Ø9H
        INT     21H
;                                       ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;
CNVNUM: CMP     BX,1119
        JG      CNVRET
        CMP     BX,Ø
        JS      CNVRET                  ;BINARY ---> NUMERIC
        MOV     CH,4
        ADD     SI,4
        MOV     AX,BX
```

```
        MOV     BL,1Ø
CNVLOOP:DIV     BL
        ADD     AH,3ØH
        MOV     DS:[SI],AH
        AND     AH,ØØH
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLEAR   DB      1BH,'[2J',1BH,'[5>h','$'
MSG     DB      1BH,'[Ø;ØH','左ボタンを離した位置は [ '
X       DB      5 DUP(' ')
        DB      ','
Y       DB      5 DUP(' ')
        DB      ' ]です。'
        DB      ØDH,ØAH,ØDH,ØAH
        DB      '何かキーを押すと終了します。',ØDH,ØAH,'$'
STOREX  DW      ?
STOREY  DW      ?
EC      DB      1BH,'[>5l','$'

DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;######### MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB       2ØØØ DUP(Ø)
MOUSEW  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      1ØØH DUP (?)
STACK   ENDS

        END     START
```

```
;###############################################################
;#                                                             #
;#     マウスカーソルの移動範囲を指定する (Ø=<X<=64Ø)            #
;#                                                             #
;###############################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX
;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[1ØØH],OFFSET INTR  ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR     ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[1ØØH],Ø            ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø            ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[1ØØH],Ø            ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø            ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH          ;描画領域色  :ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,ØF8ØØH
        MOV     DS,AX                           ;割込みセグメント
        MOV     DX,7FDØH                        ;割込みオフセット
        MOV     AH,25H
        MOV     AL,4ØH                          ;割込みタイプ:4ØH
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLEAR                 ;画面をクリア
        MOV     AH,Ø9H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                           ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                            ;割込み間隔 : 1
        INT     4ØH
```

```
;                                       ;MOUSE BIOS 作業域初期設定
        MOV     AX,Ø
        INT     4ØH
;                                       ;MICKEY/PIXEL 比を設定
        MOV     AX,15
        MOV     CX,1Ø                   ;水平方向 :1Ø
        MOV     DX,1Ø                   ;垂直方向 :1Ø
        INT     4ØH
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     4ØH
;                                       ;カーソルの移動範囲設定
H_VAL:  MOV     AX,7                    ;X座標方向を
        MOV     CX,Ø                    ;(Ø-64Ø)
        MOV     DX,64Ø
        INT     4ØH
;                                       ;カーソル位置を取得
POS:    MOV     AX,3
        INT     4ØH
        MOV     STOREX,CX               ;水平座標
        MOV     STOREY,DX               ;垂直座標
;                                       ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET X
        CALL    CNVNUM
;                                       ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET Y
        CALL    CNVNUM
;                                       ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
;                                       ;キー入力のチェック
        MOV     AH,ØBH
        INT     21H
        CMP     AL,Ø
        JE      POS
EXIT:                                   ;マウスカーソルを表示停止
        MOV     AX,2
        INT     4ØH
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     4ØH
;
        MOV     DX,OFFSET EC
        MOV     AH,Ø9H
        INT     21H
;                                       ;プログラムを終了
        MOV     AH,4CH
        INT     21H

;#########      座標の編集
CNVNUM: CMP     BX,1119
        JG      CNVRET
        CMP     BX,Ø
        JS      CNVRET
        MOV     CH,4                    ;BINARY ---> NUMERIC
        ADD     SI,4
        MOV     AX,BX
        MOV     BL,1Ø
CNVLOOP:DIV     BL
        ADD     AH,3ØH
        MOV     DS:[SI],AH
        AND     AH,ØØH
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS

;#########      データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLEAR   DB      1BH,'[2J',1BH,'[5>h','$'
MSG     DB      1BH,'[Ø;ØH','マウスカーソルの位置は [ '
X       DB      5 DUP(' ')
        DB      ','
Y       DB      5 DUP(' ')
        DB      ' ] です '
        DB      ØDH,ØAH,ØDH,ØAH
        DB      '何かキーを押すと,終了します.',ØDH,ØAH,'$'
STOREX  DW      ?
STOREY  DW      ?
EC      DB      1BH,'[>5l','$'

DATA    ENDS

;#########      中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########      GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########      MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2ØØØ DUP(Ø)
```

```
MOUSEW  ENDS
;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;###########################################################################
;#                                                                         #
;#     マウスカーソルの移動範囲を指定する  (0=<Y<=400)                     #
;#                                                                         #
;###########################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX
;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0                 ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0                 ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0                 ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1                 ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0                 ;描画領域:
        MOV     WORD PTR DS:[102H],0                 ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H               ;描画領域色 :00H
        MOV     BYTE PTR DS:[109H],0FFH              ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                                 ;割込みセグメント
        MOV     DX,7FD0H                              ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                                ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLEAR                 ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                                 ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                                  ;割込み間隔 : 1
        INT     40H
```

```
;                               ;MOUSE BIOS 作業域初期設定
        MOV     AX,Ø
        INT     4ØH
;                               ;MICKEY/PIXEL 比を設定
        MOV     AX,15
        MOV     CX,1Ø             ;水平方向 :1Ø
        MOV     DX,1Ø             ;垂直方向 :1Ø
        INT     4ØH
;                               ;マウスカーソルを表示
        MOV     AX,1
        INT     4ØH
;                               ;カーソルの移動範囲設定
H_VAL:  MOV     AX,8              ;Y座標方向を
        MOV     CX,Ø              ;(Ø-4ØØ)
        MOV     DX,4ØØ
        INT     4ØH
;                               ;カーソル位置を取得
POS:    MOV     AX,3
        INT     4ØH
        MOV     STOREX,CX         ;水平座標
        MOV     STOREY,DX         ;垂直座標
;                               ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET X
        CALL    CNVNUM
;                               ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET Y
        CALL    CNVNUM
;                               ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
;                               ;キー入力
        MOV     AH,ØBH
        INT     21H
        CMP     AL,Ø
        JE      POS
EXIT:                           ;マウスカーソルを表示停止
        MOV     AX,Ø2
        INT     4ØH
;                               ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     4ØH
;
        MOV     DX,OFFSET EC
        MOV     AH,Ø9H
        INT     21H
;                               ;プログラムを終了
        MOV     AH,4CH
        INT     21H
```

```
;#########  座標の編集
CNVNUM: CMP     BX,1119
        JG      CNVRET
        CMP     BX,Ø
        JS      CNVRET
        MOV     CH,4                      ;BINARY ---> NUMERIC
        ADD     SI,4
        MOV     AX,BX
        MOV     BL,1Ø
CNVLOOP:DIV     BL
        ADD     AH,3ØH
        MOV     DS:[SI],AH
        AND     AH,ØØH
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLEAR   DB      1BH,'[2J',1BH,'[5>h','$'
MSG     DB      1BH,'[Ø;ØH','マウスカーソルの位置は [ '
X       DB      5 DUP(' ')
        DB      ','
Y       DB      5 DUP(' ')
        DB      '] です'
        DB      ØDH,ØAH,ØDH,ØAH
        DB      '何かキーを押すと終了します.',ØDH,ØAH,'$'
STOREX  DW      ?
STOREY  DW      ?
EC      DB      1BH,'[>5l','$'
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)                ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                ;パラメータリスト
GWORK   DB      512 DUP(?)                ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########  MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2ØØØ DUP(Ø)
```

```
MOUSEW  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                                #
;#      マウスカーソルの形状を指定する                            #
;#                                                                #
;##################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX

;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR  ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR     ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0            ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0            ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0            ;描画領域:
        MOV     WORD PTR DS:[102H],0            ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H          ;描画領域色 :00H
        MOV     BYTE PTR DS:[109H],0FFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                           ;割込みセグメント
        MOV     DX,7FD0H                        ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                          ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS                   ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                           ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                            ;割込み間隔 :1
        INT     40H
```

```
;                                   ;MOUSE BIOS 作業域初期設定
        MOV     AX,0
        INT     40H
;                                   ;カーソルの形状を設定
        PUSH    ES
        MOV     BX,0                  ;中心点のX座標:0
        MOV     CX,0                  ;中心点のY座標:0
        MOV     SI,1010H              ;X:16bit,Y:16bit
        MOV     AX,SEG CURSOR         ;カーソルパターン
        MOV     ES,AX
        MOV     DX,OFFSET CURSOR
        MOV     AX,9
        INT     40H
        POP     ES
;                                   ;表示プレーンを設定
        MOV     AX,16
        MOV     BX,00AFH
        INT     40H
;                                   ;MICKEY/PIXEL 比を設定
        MOV     AX,15
        MOV     CX,10                 ;水平方向 :10
        MOV     DX,10                 ;垂直方向 :10
        INT     40H
;                                   ;マウスカーソルを表示
        MOV     AX,1
        INT     40H
;                                   ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,08H              ;キー入力待ち
        INT     21H
;                                   ;マウスカーソルの表示停止
        MOV     AX,2
        INT     40H                 ;マウス割込みを STOP
;
        MOV     AX,18
        MOV     BX,1
        INT     40H
EXIT:   MOV     AH,4CH              ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG     DB      '何かキーを押すと,マウスカーソルが消えます.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########  MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2000 DUP(0)
CURSOR  DB      11000000B,00000111B
        DB      11110100B,00101111B
        DB      11111010B,01011111B
        DB      11111010B,01011111B
        DB      11111101B,10111111B
        DB      11111101B,10111111B
        DB      11111111B,11111111B
        DB      01111110B,01111110B
        DB      01111110B,01111110B
        DB      01111110B,01111110B
        DB      00011111B,11111000B
        DB      01111111B,11111110B
        DB      01111110B,01111110B
        DB      01111100B,00111110B
        DB      00111000B,00011100B
        DB      00011000B,00011000B
MOUSEW  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                               #
;#      マウスの移動距離を取得する                                 #
;#                                                               #
;##################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW              ;作業領域セグメントを設定
        MOV     DS,AX
;                                          ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                          ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[1ØØH],Ø         ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø         ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø         ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1         ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                          ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[1ØØH],Ø         ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø         ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH       ;描画領域色  :ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH      ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                          ;割込みベクタをセット
        MOV     AX,ØF8ØØH
        MOV     DS,AX                        ;割込みセグメント
        MOV     DX,7FDØH                     ;割込みオフセット
        MOV     AH,25H
        MOV     AL,4ØH                       ;割込みタイプ:4ØH
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS              ;画面をクリア
        MOV     AH,Ø9H
        INT     21H
;                                          ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                        ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                         ;割込み間隔 : 1
        INT     4ØH
;                                          ;MOUSE BIOS 作業域初期設定
        MOV     AX,Ø
        INT     4ØH
;                                          ;マウスカーソルを表示
        MOV     AX,1
        INT     4ØH
;                                          ;MICKEY/PIXEL 比を設定
        MOV     AX,15
        MOV     CX,1Ø                        ;水平方向 :1Ø
        MOV     DX,1Ø                        ;垂直方向 :1Ø
        INT     4ØH
;                                          ;メッセージを表示
        MOV     AH,Ø9H
        MOV     DX,OFFSET MSG1
        INT     21H
;                                          ;キー入力のチェック
POS:    MOV     AH,ØBH
        INT     21H
        CMP     AL,Ø
        JNE     EXIT
;                                          ;ボタン押下位置を取得
        MOV     AX,5
        MOV     BX,Ø                         ;左ボタンを取得
        INT     4ØH
        TEST    AX,1                         ;左ボタン押下 ?
        JZ      POS
        MOV     STOREX,CX                    ;水平座標
        MOV     STOREY,DX                    ;垂直座標
;                                          ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET MSG2X
        CALL    CNVNUM
;                                          ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET MSG2Y
        CALL    CNVNUM
;                                          ;メッセージを表示
        MOV     AH,Ø9H
        MOV     DX,OFFSET MSG2
        INT     21H
;                                          ;移動距離取得(1回目)
        MOV     AX,11
        INT     4ØH
;                                          ;キー入力のチェック
UP:     MOV     AH,ØBH
        INT     21H
        CMP     AL,Ø
        JNE     EXIT
;                                          ;ボタン解放位置を取得
        MOV     AX,6
        MOV     BX,Ø                         ;左ボタンをオーダー
        INT     4ØH
        TEST    AX,1                         ;左ボタン押下 ?
```

```
        JNZ     UP
;                                       ;移動距離取得
MOTION: MOV     AX,11
        INT     4ØH
        MOV     STOREX,CX                 ;水平座標
        MOV     STOREY,DX                 ;垂直座標
;                                       ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET MSG3X
        CALL    CNVNUM
;                                       ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET MSG3Y
        CALL    CNVNUM
;                                       ;メッセージを表示
        MOV     AH,Ø9H
        MOV     DX,OFFSET MSG3
        INT     21H
        JMP     POS
;
EXIT:                                   ;マウスカーソルの表示停止
        MOV     AX,2
        INT     4ØH
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     4ØH
;
        MOV     AH,9
        MOV     DX,OFFSET EC
        INT     21H
;                                       ;プログラムを終了
        MOV     AH,4CH
        INT     21H

;#########  座標の編集
CNVNUM: CMP     BX,Ø
        JS      CNV_M
        MOV     BYTE PTR DS:[SI],' '
        MOV     AX,BX
        JMP     CNV_N
CNV_M:  MOV     BYTE PTR DS:[SI],'-'
        MOV     AX,BX
        NEG     AX
CNV_N:  XOR     DX,DX
        MOV     CH,5                    ;BINARY ---> NUMERIC
        ADD     SI,5
        MOV     BP,1Ø
CNVLOOP:DIV     BP
        ADD     DL,3ØH
        MOV     DS:[SI],DL
        XOR     DX,DX
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS
;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG1    DB      1BH,'[5>h'
        DB      '左ボタンを押しながらマウスを移動し,'
        DB      'ボタンを離してください.',ØDH,ØAH,ØDH,ØAH
        DB      '何かキーを押すと終了します.',ØDH,ØAH,'$'
MSG2    DB      1BH,'[2J',1BH,'[5>h','ボタンを押した位置は [ '
MSG2X   DB      6 DUP(' ')
        DB      ','
MSG2Y   DB      6 DUP(' ')
        DB      '] です.',ØDH,ØAH,ØDH,ØAH,'$'
MSG3    DB      '移動した距離は [ '
MSG3X   DB      6 DUP(' ')
        DB      ','
MSG3Y   DB      6 DUP(' ')
        DB      '] です',ØDH,ØAH,'$'
STOREX  DW      ?
STOREY  DW      ?
EC      DB      1BH,'[>5l','$'
DATA    ENDS
;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS
;#########  GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########  MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2ØØØ DUP(Ø)
MOUSEW  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      1ØØH DUP (?)
STACK   ENDS
        END     START
```

```
;###################################################################
;#                                                                 #
;#     マウスによるユーザ定義サブルーチンへの分岐                        #
;#                                                                 #
;###################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW           ;作業領域セグメントを設定
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                       ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[1ØØH],Ø            ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø            ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø            ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1            ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[1ØØH],Ø            ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø            ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH          ;描画領域色  :ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH         ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;割込みベクタをセット
        MOV     AX,ØF8ØØH
        MOV     DS,AX                           ;割込みセグメント
        MOV     DX,7FDØH                        ;割込みオフセット
        MOV     AH,25H
        MOV     AL,4ØH                          ;割込みタイプ:4ØH
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLEAR         ;画面をクリア
        MOV     AH,Ø9H
        INT     21H
;                                       ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                           ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                            ;割込み間隔 : 1
        INT     4ØH
;                                       ;MOUSE BIOS 作業域初期設定
        MOV     AX,Ø
        INT     4ØH
;                                       ;MICKEY/PIXEL 比を設定
        MOV     AX,15
        MOV     CX,1Ø                           ;水平方向 :1Ø
        MOV     DX,1Ø                           ;垂直方向 :1Ø
        INT     4ØH
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     4ØH
;                                       ;ユーザー定義サブルーチン
        PUSH    ES
        MOV     AX,SEG USRSUB                   ;ルーチンセグメント
        MOV     ES,AX
        MOV     DX,Ø                            ;ルーチンオフセット
        MOV     CX,2                            ;条件:左ボタン押下
        MOV     AX,12
        INT     4ØH
        POP     ES
;                                       ;カーソル位置を取得
POS:    MOV     AX,3
        INT     4ØH
        MOV     STOREX,CX                       ;水平座標
        MOV     STOREY,DX                       ;垂直座標
;                                       ;水平座標を編集
        MOV     BX,STOREX
        MOV     SI,OFFSET X
        CALL    CNVNUM
                                        ;垂直座標を編集
        MOV     BX,STOREY
        MOV     SI,OFFSET Y
        CALL    CNVNUM
;                                       ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
                                        ;割込みのチェック
        MOV     AL,WARI
        CMP     AL,49H
        JNE     NO_INT
        MOV     DX,OFFSET INT_MSG
        JMP     DSP_MSG
NO_INT: MOV     DX,OFFSET NINT_MSG
DSP_MSG:MOV     AH,9
        INT     21H
        MOV     BYTE PTR WARI,3ØH
;                                       ;キー入力
CHKKEY: MOV     AH,ØBH
        INT     21H
        CMP     AL,Ø
        JE      POS
EXIT:                                   ;マウスカーソルを表示停止
```

```
        MOV     AX,02
        INT     40H                     ;マウス割込みを STOP
;
        MOV     AX,18
        MOV     BX,1
        INT     40H
;
        MOV     DX,OFFSET EC
        MOV     AH,09H
        INT     21H
;
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H

;#########  座標の編集
CNVNUM: CMP     BX,1119
        JG      CNVRET
        CMP     BX,0
        JS      CNVRET
        MOV     CH,4                    ;BINARY ---> NUMERIC
        ADD     SI,4
        MOV     AX,BX
        MOV     BL,10
CNVLOOP:DIV     BL
        ADD     AH,30H
        MOV     DS:[SI],AH
        AND     AH,00H
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLEAR   DB      1BH,'[2J',1BH,'[5>h','$'
MSG     DB      1BH,'[0;0H','マウスカーソルの位置は ['
X       DB      5 DUP(' ')
        DB      ','
Y       DB      5 DUP(' ')
        DB      ' ] です',0DH,0AH,0DH,0AH
        DB      '左ボタンを押してみて下さい',0DH,0AH
        DB      0DH,0AH,0DH,0AH
        DB      '何かキーを押すと終了します.',0DH,0AH,'$'
STOREX  DW      ?
STOREY  DW      ?
EC      DB      1BH,'[>5l','$'
WARI    DB      30H
INT_MSG DB      1BH,'[10;20H',07H,'Interrupt !!',0DH,0AH,'$'
NINT_MSG DB     1BH,'[10;20H','              ',0DH,0AH,'$'
DATA    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;#########  マウス割込み用ユーザー定義サブルーチン
USRSUB  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:USRSUB,DS:DATA
MICKSUB PROC    FAR
        MOV     AX,DATA
        MOV     DS,AX
        MOV     BYTE PTR WARI,49H
        RET
MICKSUB ENDP
USRSUB  ENDS

;#########  GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;#########  MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2000 DUP(0)
MOUSEW  ENDS

;#########  スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;#####################################################################
;#                                                                   #
;#       M I C K E Y / P I X E L 比を設定する                          #
;#                                                                   #
;#####################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW           ;作業領域セグメントを設定
        MOV     DS,AX
;                                       ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR    ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                       ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0    ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0    ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0    ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1    ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                       ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0    ;描画領域:
        MOV     WORD PTR DS:[102H],0    ;   (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H  ;描画領域色  :00H
        MOV     BYTE PTR DS:[109H],0FFH ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                       ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                   ;割込みセグメント
        MOV     DX,7FD0H                ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                  ;割込みタイプ:40H
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS           ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                       ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                   ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                    ;割込み間隔 : 1
        INT     40H
;
        MOV     AX,0                    ;MOUSE BIOS 作業域初期設定
        INT     40H
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     40H
;                                       ;MICKEY/PIXEL 比設定
        MOV     AX,15
        MOV     CX,100H                 ;水平方向:100
        MOV     DX,100H                 ;垂直方向:100
        INT     40H
;                                       ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,09H
        INT     21H
        MOV     AH,08H                  ;キー入力待ち
        INT     21H
;                                       ;マウスカーソルの表示停止
        MOV     AX,2
        INT     40H
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     40H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG     DB      '何かキーを押すと,マウスカーソルが消えます.'
        DB      0DH,0AH,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;######### MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
```

```
MW      DB      2000 DUP(0)
MOUSEW  ENDS

;######## スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;################################################################
;#                                                              #
;#        Mouse カーソルの表示画面を設定する                    #
;#                                                              #
;################################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX
;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[100H],OFFSET INTR  ;中断処理オフセット
        MOV     WORD PTR DS:[102H],SEG INTR     ;中断処理セグメント
        MOV     AH,0
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[100H],0                ;表示モード:カラー
        MOV     BYTE PTR DS:[101H],0                ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[102H],0                ;アクティブページ  :0
        MOV     BYTE PTR DS:[103H],1                ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[100H],0                ;描画領域:
        MOV     WORD PTR DS:[102H],0                ;  (0,0)-(1119,749)
        MOV     WORD PTR DS:[104H],1119
        MOV     WORD PTR DS:[106H],749
        MOV     BYTE PTR DS:[108H],00H              ;描画領域色 :00H
        MOV     BYTE PTR DS:[109H],0FFH             ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,0F800H
        MOV     DS,AX                               ;割込みセグメント
        MOV     DX,7FD0H                            ;割込みオフセット
        MOV     AH,25H
        MOV     AL,40H                              ;割込みタイプ:40H
        INT     21H

        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS                   ;画面をクリア
        MOV     AH,09H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                               ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                                ;割込み間隔 : 1
        INT     40H
```

```
;       MOV     AX,0                    ;MOUSE BIOS 作業域初期設定
        INT     40H
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     40H
;                                       ;表示プレーンを設定
        MOV     AX,16
        MOV     BX,00A2H                ;2画面へ表示
        INT     40H
;                                       ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     BX,AX                   ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                    ;割込み間隔 : 1
        INT     40H
;                                       ;マウスカーソルを表示
        MOV     AX,1
        INT     40H
;                                       ;メッセージを表示
        MOV     DX,OFFSET MSG2
        MOV     AH,09H
        INT     21H
        MOV     AH,08H                  ;キー入力待ち
        INT     21H
;                                       ;マウスカーソルの表示停止
        MOV     AX,2
        INT     40H
;                                       ;マウス割込みを STOP
        MOV     AX,18
        MOV     BX,1
        INT     40H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG1    DB      '何かキーを押すと,表示画面が変わります.',0DH,0AH,'$'
MSG2    DB      0DH,0AH,'何かキーを押すと,終了します.',0DH,0AH,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)              ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)              ;パラメータリスト
GWORK   DB      512 DUP(?)              ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;######### MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2000 DUP(0)
MOUSEW  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      100H DUP (?)
STACK   ENDS

        END     START
```

```
;#############################################################
;#                                                           #
;#     マウスの割込みを不許可にする                              #
;#                                                           #
;#############################################################
MOUSE   GROUP   CODE,DATA,MOUSEW,GRAPHW,INTRPT
CODE    SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:CODE,DS:GRAPHW,ES:MOUSEW,SS:STACK
START:
        MOV     AX,SEG GRAPHW                   ;作業領域セグメントを設定
        MOV     DS,AX
;                                               ;GRAPH BIOS の初期設定
GINIT:  MOV     WORD PTR DS:[1ØØH],OFFSET INTR ;中断処理オフセット
        MOV     WORD PTR DS:[1Ø2H],SEG INTR    ;中断処理セグメント
        MOV     AH,Ø
        INT     1DH
;                                               ;グラフ画面のモードを設定
GSCREEN:MOV     BYTE PTR DS:[1ØØH],Ø               ;表示モード:カラー
        MOV     BYTE PTR DS:[1Ø1H],Ø               ;表示SW:表示あり,普通
        MOV     BYTE PTR DS:[1Ø2H],Ø               ;アクティブページ  :Ø
        MOV     BYTE PTR DS:[1Ø3H],1               ;ディスプレイページ:1
        MOV     AH,1
        INT     1DH
;                                               ;描画領域を設定し,塗る
GVIEW:  MOV     WORD PTR DS:[1ØØH],Ø               ;描画領域:
        MOV     WORD PTR DS:[1Ø2H],Ø               ;  (Ø,Ø)-(1119,749)
        MOV     WORD PTR DS:[1Ø4H],1119
        MOV     WORD PTR DS:[1Ø6H],749
        MOV     BYTE PTR DS:[1Ø8H],ØØH             ;描画領域色 :ØØH
        MOV     BYTE PTR DS:[1Ø9H],ØFFH            ;外枠は描かない
        MOV     AH,2
        INT     1DH
;                                               ;割込みベクタをセット
        MOV     AX,ØF8ØØH
        MOV     DS,AX                              ;割込みセグメント
        MOV     DX,7FDØH                           ;割込みオフセット
        MOV     AH,25H
        MOV     AL,4ØH                             ;割込みタイプ:4ØH
        INT     21H
;
        ASSUME  DS:DATA
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
        MOV     DX,OFFSET CLS                   ;画面をクリア
        MOV     AH,Ø9H
        INT     21H
;                                               ;マウス割込み間隔設定
        MOV     AX,SEG MOUSEW
        MOV     ES,AX
        MOV     BX,AX                              ;MOUSE BIOS 作業領域
        MOV     AX,17
        MOV     CX,1                               ;割込み間隔 : 1
        INT     4ØH
        MOV     AX,Ø                            ;MOUSE BIOS 作業域初期設定
        INT     4ØH
        MOV     AX,1                            ;マウスカーソルを表示
        INT     4ØH
;                                               ;メッセージを表示
        MOV     DX,OFFSET MSG
        MOV     AH,Ø9H
        INT     21H
        MOV     AH,Ø8H                          ;キー入力待ち
        INT     21H
;                                               ;マウスカーソルの表示停止
        MOV     AX,2
        INT     4ØH
        MOV     AX,18                           ;マウス割込みを STOP
        MOV     BX,1
        INT     4ØH
EXIT:
        MOV     AH,4CH                          ;プログラムを終了
        INT     21H
CODE    ENDS

;######### データ領域
DATA    SEGMENT PARA PUBLIC 'MOUSE'
CLS     DB      1BH,'[2J','$'
MSG     DB      '何かキーを押すと,マウスカーソルが消えます.'
        DB      ØDH,ØAH,'$'
DATA    ENDS

;######### 中断処理ルーチン
INTRPT  SEGMENT PARA PUBLIC 'MOUSE'
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        RET
INTR    ENDP
INTRPT  ENDS

;######### GRAPH BIOS 使用作業領域
GRAPHW  SEGMENT PARA PUBLIC 'MOUSE'
GBIOS   DB      256 DUP(?)                      ;GRAPH BIOS UCW
PARALST DB      128 DUP(?)                      ;パラメータリスト
GWORK   DB      512 DUP(?)                      ;GRAPH BIOS 作業領域
GRAPHW  ENDS

;######### MOUSE BIOS 使用作業領域
MOUSEW  SEGMENT PARA PUBLIC 'MOUSE'
MW      DB      2ØØØ DUP(Ø)
MOUSEW  ENDS

;######### スタック領域
STACK   SEGMENT PARA STACK 'STACK'
        DW      1ØØH DUP (?)
STACK   ENDS

        END     START
```

# 第7章

# プリンタBIOS

## ● プリンタ BIOS 機能一覧

| AHレジスタ | 機　　能 |
|---|---|
| 10H | 初期化 |
| 11H | データの出力(Wait モード) |
| 12H | ステータスの取得 |
| 14H | データの出力(No Wait モード) |
| 15H | データの出力(チェックレスモード) |
| 16H | 初期化(Wait 時間設定) |
| 30H | データの出力(複数バイト) |

## 7.1　プリンタ BIOS 共通情報

### (1) 概要

PC-98XL/XA のプリンタ BIOS は，セントロニクスインターフェイスプリンタに関する基本的な I/O をサポートするものである．

①プリンタの初期化を行う．

②データ出力処理を行う．

a ) Wait モード

I/O 終了まで BIOS 内でループして処理を待つ．シングルタスク用モードである．

b ) No Wait モード

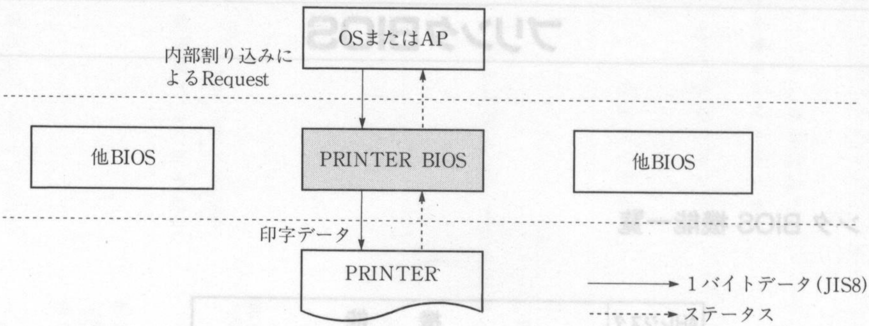
Wait モードのループの前に INT 1FH の Wait(Only)ファンクションをコールする．これによって BIOS 内でループ動作を行う可能性が減り，CPU の使用効率が向上する．マルチタスク用モードである．

c ) チェックレスモード

与えられたデータを無条件にプリンタへ送出する．

③センスステータス処理を行う．

**(2) 本ルーチンの位置付け**




**(3) 注意事項**

本BIOSはプリンタからのハードウェア割り込みを使用しない.

**(4) 制限事項**

本BIOSはBUSYチェック方式のため，BUSY中DC1コードを送出できない．そのためDC1/DC3の使用は出来ない.

**(5) ハードウェアインターフェイス**

・8255Aを使用.

・フルセントロニクスインターフェイスで動作するプリンタを接続可能.

**(6) システム共通域**

| システム共通域<br>シンボル名・絶対番地 | 用　　途 |
|---|---|
| PR_TIME<br>(41CH, 41DH) | プリンタのタイムアウト値を設定する<br>0000H：無限<br>0001H～FFFFH(10～655350msec) |

## 7.2　初期化

### (1) 機能

μPD8255A のモード設定（ポートA：出力，ポートB：入力，ポートC：出力），ステータス情報の初期化を行う．プリンタ BUSY 時の Wait 時間は 4Sec となる．

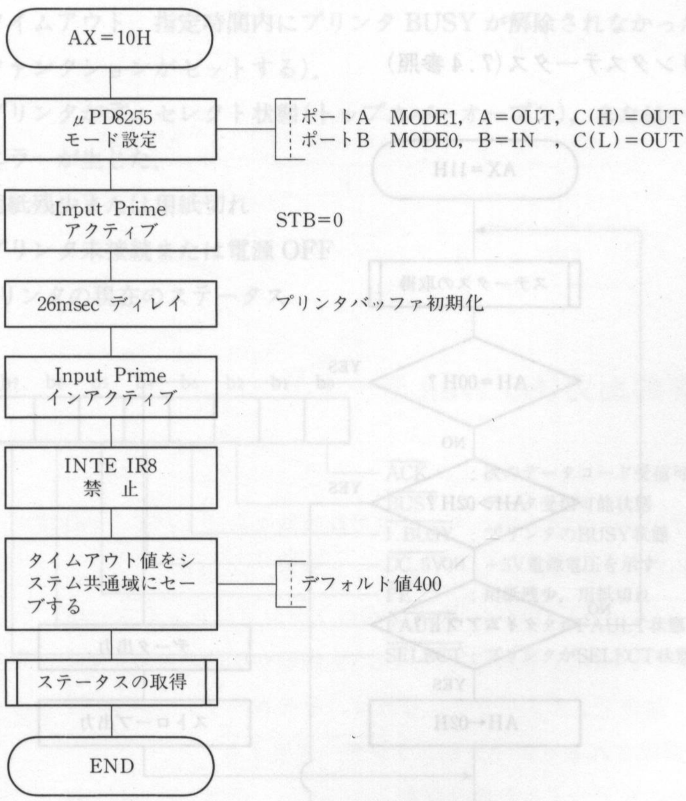
### (2) 入力

・内部割り込みコード←1AH

・AH ←10H（BIOS コマンド識別コード）

### (3) 出力

・AX ← プリンタステータス（7.4 参照）



## 7.3　データの出力(Wait モード)

### (1) 機能

プリンタのステータスを読み出して，受信可能状態になるまでループ処理をして待つ．プリンタ初期化コマンドで指定した待ち時間を経過するとタイムアウトのリターンコードをセットする．受信可能になったら1バイトの印字データをプリンタに送出して，送出済みのリターンコードをセットする．

### (2) 入力

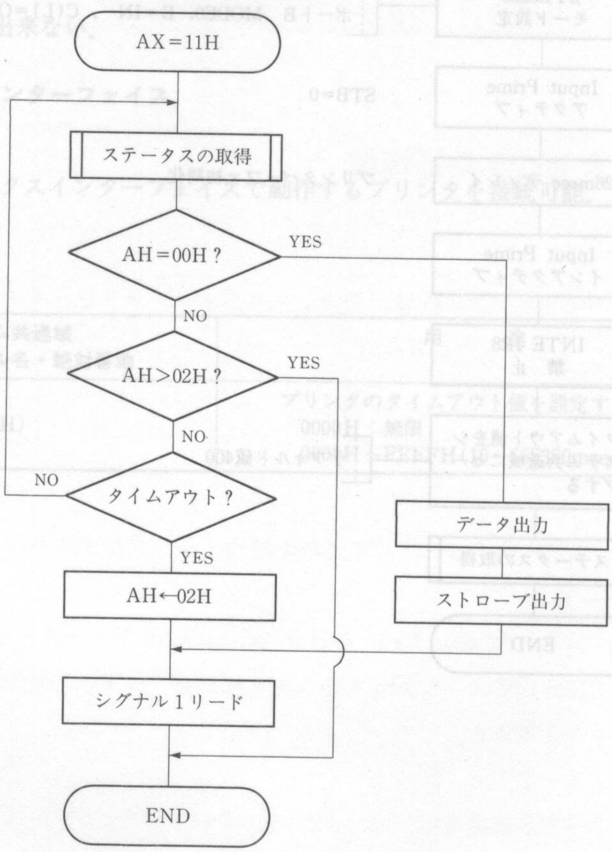
・内部割り込みコード←1AH

・AH ←11H(BIOS コマンド識別コード)

・AL ←出力する1バイトのデータ(JIS8 ビットコード)

### (3) 出力

・AX ←プリンタステータス(7.4参照)



# 7.4 ステータスの取得

## (1) 機能

現在のプリンタのステータス情報の通知を要求し，その通知を受ける．

## (2) 入力

・内部割り込みコード← 1AH

・AH ← 12H (BIOS コマンド識別コード)

## (3) 出力

・AH ← ステータス情報

00H：プリンタがデータ受信可能

01H：プリンタ BUSY 状態

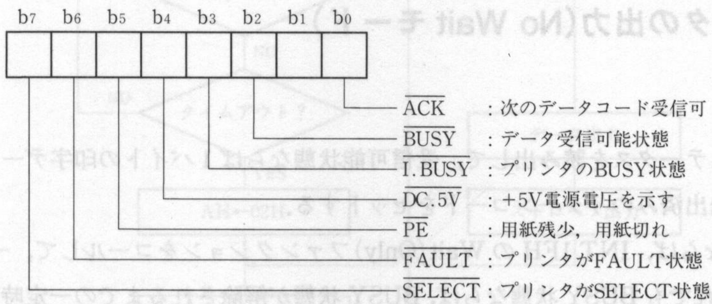
02H：タイムアウト．指定時間内にプリンタ BUSY が解除されなかった(本機能以外のファンクションがセットする)．

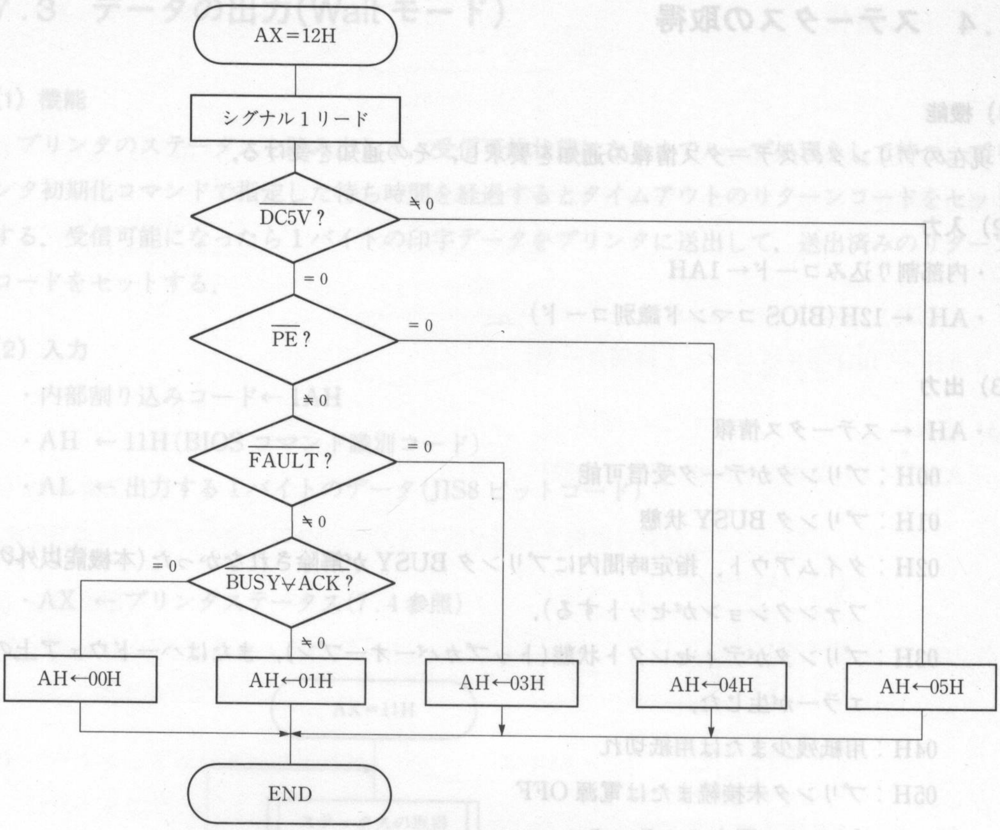
03H：プリンタがディセレクト状態(トップカバーオープン)，またはハードウェア上のエラーが生じた．

04H：用紙残少または用紙切れ

05H：プリンタ未接続または電源 OFF

・AL ← プリンタの現在のステータス






## 7.5　データの出力(No Wait モード)

### (1) 機能

プリンタのステータスを読み出して，受信可能状態ならば１バイトの印字データをプリンタに送出して，送出済みのリターンコードをセットする．

BUSY 状態ならば，INT 1FH の Wait(Only)ファンクションをコールして，一時的に CPU を解放する．それでも BUSY 状態ならば，BUSY 状態が解除されるまでの一定時間ソフトウェアループを行う．途中でタイムアウトになれば，その旨のリターンコードをセットする．

BUSY 状態が解除されるとデータをプリンタへ送出する．

最後にプリンタのステータスを読み出す．

### (2) 入力

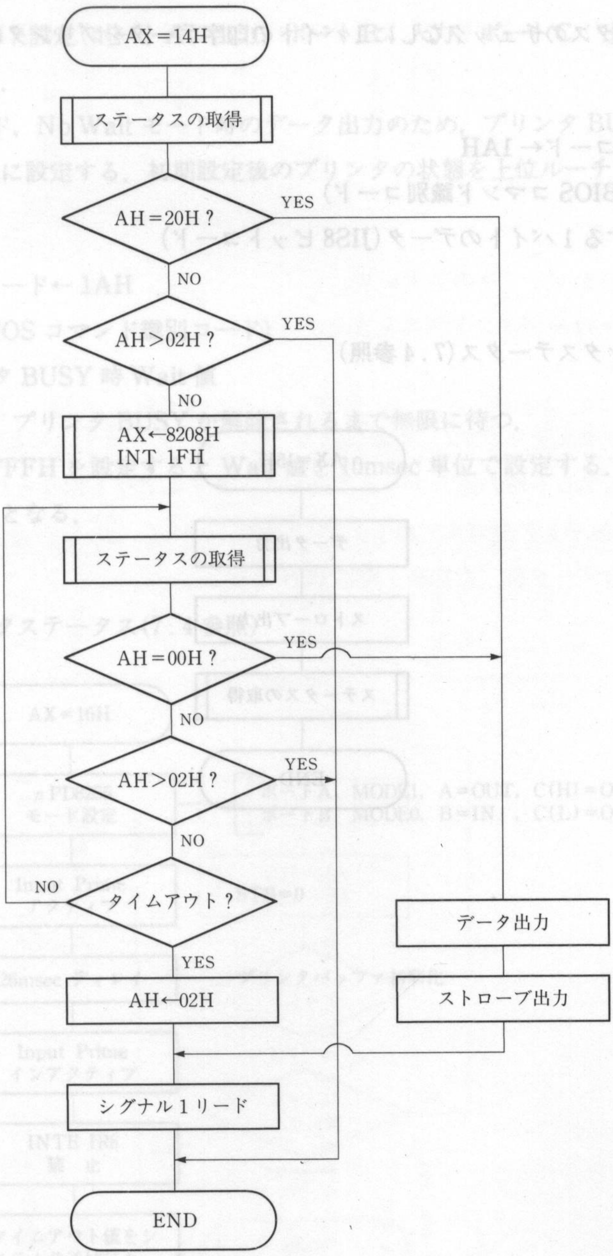
・内部割り込みコード←1AH

・AH ←14H(BIOS コマンド識別コード)

・AL ←出力する１バイトのデータ(JIS8 ビットコード)

(3) 出力

・AX ←プリンタステータス(7.4参照)



## 7.6　データの出力(チェックレスモード)

### (1) 機能

プリンタステータスのチェックなしに1バイトの印字データをプリンタに送出する.

### (2) 入力

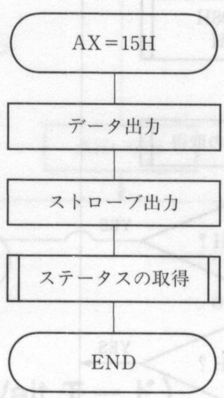
・内部割り込みコード←1AH

・AH　←15H(BIOS コマンド識別コード)

・AL　←出力する1バイトのデータ(JIS8 ビットコード)

### (3) 出力

・AX　←プリンタステータス(7.4参照)



# 7.7　初期化(Wait 時間設定)

## (1) 機能

μPD8255A のモード設定(ポートA：出力，ポートB：入力，ポートC：出力)，ステータス情報の初期化を行う．

また，Wait モード，No Wait モード時のデータ出力のため，プリンタ BUSY 時の Wait 時間をシステム共通域に設定する．初期設定後のプリンタの状態を上位ルーチンに返す．

## (2) 入力

・内部割り込みコード← 1AH

・AH ← 16H(BIOS コマンド識別コード)

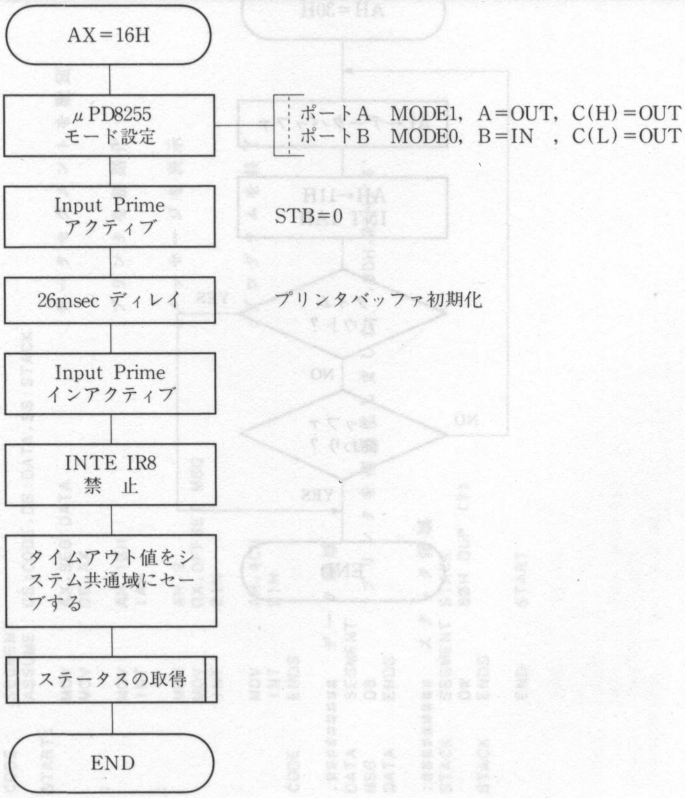
・CX ← プリンタ BUSY 時 Wait 値

0000H の時，プリンタ BUSY が解除されるまで無限に待つ．

0001H～0FFFFH を設定すると Wait 値を 10msec 単位で設定する．0FFFFH の時，655350msec となる．

## (3) 出力

・AX ← プリンタステータス(7.4 参照)



# 7.8　データの出力(複数バイト)

### (1) 機能

指定したデータバッファ上の指定した長さの印字データをプリンタに送出する.

### (2) 入力

・内部割り込みコード←1AH

・AH ← 30H(BIOS コマンド識別コード)

・CX ←データ長

・ES/BX ←データバッファのアドレス

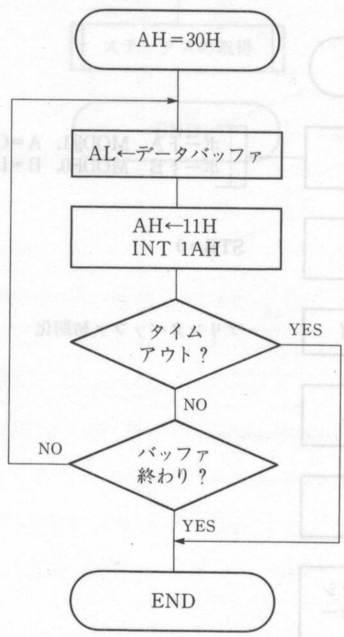
注：データバッファはセグメント境界をまたがってはいけない.

### (3) 出力

・AX ←プリンタステータス(7.4 参照)

・BX ←タイムアウト時の出力データアドレス. タイムアウトになったデータバッファ上のオフセットアドレスで示す.

・CX ←未出力データ長. 正常終了時は 00H となる.



●**サンプルリスト**：本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;###################################################################
;#                                                                 #
;#        プリンタを初期化する                                     #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,10H                          ;プリンタを初期化
        INT     1AH
;
        MOV     AH,9                            ;メッセージを表示
        MOV     DX,OFFSET MSG
        INT     21H
;
        MOV     AH,4CH                          ;プログラムを終了
        INT     21H
CODE    ENDS

;#########   データ領域
DATA    SEGMENT
MSG     DB      'プリンタを初期化しました.',0DH,0AH,'$'
DATA    ENDS

;#########   スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;###################################################################
;#                                                                 #
;#        Waitモードでデータを出力する                             #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK,DS:DATA
START:
        MOV     AX,SEG DATA                     ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,10H                          ;プリンタを初期化
        INT     1AH
;
        MOV     CX,10H                          ;10回繰返し
PRNT:   MOV     AL,'X'                          ;{
        CALL    OUTPUT                          ;  文字"X"を出力
        LOOP    PRNT                            ;}
;
        MOV     AL,0DH                          ;CR をプリンタへ出力
        CALL    OUTPUT
        MOV     AL,0AH                          ;LF をプリンタへ出力
        CALL    OUTPUT
;
        MOV     AH,4CH                          ;プログラムを終了
        INT     21H

;######### 1 文字出力
OUTPUT: MOV     AH,11H                          ;WAITモードデータ出力
        INT     1AH
        CMP     AH,00H
        JE      OUT_E                           ;AT NORMAL   :RETURN
        CMP     AH,02H
        JE      OUTPUT                          ;AT TIMEOUT  :RETRY
MSGOUT: MOV     AH,09H                          ;AT TROUBLE  :MSG.OUT
        MOV     DX,OFFSET MSG
        INT     21H
        MOV     AH,01H
        INT     21H
        CMP     AL,'R'
        JE      OUTPUT
        CMP     AL,'r'
        JE      OUTPUT
        CMP     AL,'I'
        JE      OUT_E
        CMP     AL,'i'
        JE      OUT_E
        CMP     AL,'A'
        JE      ABORT
        CMP     AL,'a'
        JNE     MSGOUT
ABORT:  MOV     AH,4CH                          ;プログラムを終了
        INT     21H
```

```
OUT_E:  RET

CODE    ENDS

;#########  データ領域
DATA    SEGMENT
MSG     DB      ØDH,ØAH,'プリンタの準備ができていません.'
        DB      ØDH,ØAH,'中止<A>,もう一度<R>,無視<I>? ','$'
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;###################################################################
;#                                                                 #
;#      プリンタの状態を表示する                                   #
;#                                                                 #
;###################################################################
CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメントを設定
        MOV     DS,AX
;
        MOV     AH,12H                  ;プリンタステータス取得
        INT     1AH
;
DC_5V:  TEST    AL,1ØH                  ;電源 OK ?
        JZ      PE
        MOV     DX,OFFSET M_DC5V
        JMP     DISP
PE:     TEST    AL,2ØH                  ;用紙切れ ?
        JNZ     SEL
        MOV     DX,OFFSET M_PE
        JMP     DISP
SEL:    TEST    AL,8ØH                  ;SELECT 状態 ?
        JZ      FOULT
        MOV     DX,OFFSET M_DSEL
        JMP     DISP
FOULT:  TEST    AL,4ØH                  ;FOULT ?
        JNZ     BUSY
        MOV     DX,OFFSET M_FOULT
        JMP     DISP
BUSY:   TEST    AL,Ø4H                  ;プリンタBUSY ?
        JNZ     READY
        MOV     DX,OFFSET M_BUSY
        JMP     DISP
READY:  MOV     DX,OFFSET M_READY
;
DISP:   MOV     AH,9                    ;メッセージを表示
        INT     21H
EXIT:
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT
M_FOULT DB      'プリンタエラーです.',ØAH,ØDH,'$'
M_BUSY  DB      'プリンタは BUSY 状態です.',ØAH,ØDH,'$'
M_PE    DB      '用紙切れです.',ØAH,ØDH,'$'
M_DSEL  DB      'プリンタは DESELECT 状態です.',ØAH,ØDH,'$'
M_READY DB      'プリンタは印字可能です.',ØAH,ØDH,'$'
M_DC5V  DB      '電源 OFF か,接続不良です.',ØAH,ØDH,'$'
DATA    ENDS
```

```
;#########  スタック領域
STACK     SEGMENT STACK
          DW      8ØH DUP (?)
STACK     ENDS

          END     START
```

```
;##########################################################
;#                                                        #
;#      N o  W a i t モードでデータを出力する             #
;#                                                        #
;##########################################################
CODE      SEGMENT
          ASSUME  CS:CODE,SS:STACK
START:
          MOV     AH,1ØH                   ;プリンタを初期化
          INT     1AH
;                                          ;No waitモードで"A"出力
LOOP:     MOV     AH,12H                       ;センスステータス
          MOV     AL,'A'
          INT     1AH
;                                          ;画面に"S"を表示
          MOV     AH,2
          MOV     DL,'S'
          INT     21H
;                                          ;キー入力をチェック
          MOV     AH,ØBH
          INT     21H
          CMP     AL,Ø
          JE      LOOP
;
          MOV     AH,4CH
          INT     21H
;
CODE      ENDS

;#########  スタック領域
STACK     SEGMENT STACK
          DW      8ØH DUP (?)
STACK     ENDS
          END     START
```

```
;##############################################################
;#                                                            #
;#    ステータスのチェックをしないでデータを出力する            #
;#                                                            #
;##############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
START:
        MOV     CX,80                   ;80 文字分繰返し
;                                       ;{
A:      MOV     BL,'A'                  ;  "A"を出力
        CALL    PRNT
        LOOP    A                       ;}
;                                       ;LF を出力
        MOV     BL,0DH
        CALL    PRNT
;                                       ;CR を出力
        MOV     BL,0AH
        CALL    PRNT
EXIT:                                   ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;
PRNT:   MOV     AH,12H                  ;センスステータス
        INT     1AH
        CMP     AH,00H                  ;受信可:出力
        JE      PR_CH
        CMP     AH,01H                  ;BUSY  :リトライ
        JE      PRNT
        JMP     PR_RET                  ;ERROR/用紙切れ:終了
;
PR_CH:  MOV     AH,15H                  ;文字を出力
        MOV     AL,BL
        INT     1AH
PR_RET: RET

CODE    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS
        END     START
```

```
;##############################################################
;#                                                            #
;#    Wait時間設定（プリンタの初期化）                          #
;#                                                            #
;##############################################################
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
;                                       ;Wait時間設定/初期化
START:  MOV     AH,16H
        MOV     CX,500
        INT     1AH
;                                       ;プログラムを終了する
        MOV     AH,4CH
        INT     21H
;
CODE    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;####################################################################
;#                                                                  #
;#        複数バイトデータの出力                                    #
;#                                                                  #
;####################################################################
CODE      SEGMENT
          ASSUME   CS:CODE,DS:DATA,SS:STACK
START:
          MOV      AX,SEG DATA                 ;データセグメントを設定
          MOV      DS,AX
;
SENSE:    MOV      AH,12H                         ;センスステータス
          INT      1AH
          CMP      AH,ØØH                         ;受信可:出力
          JE       PRNT
          CMP      AH,Ø1H                         ;BUSY  :リトライ
          JE       SENSE
          JMP      EXIT                           ;ERROR/用紙切れ:終了
;                                              ;複数バイトデータの出力
PRNT:     MOV      AH,3ØH
          MOV      BX,SEG BUFFER
          MOV      ES,BX
          MOV      BX,OFFSET BUFFER
          MOV      CX,23
          INT      1AH
EXIT:                                          ;プログラムを終了
          MOV      AH,4CH
          INT      21H
CODE      ENDS

;#########   データ領域
DATA      SEGMENT
BUFFER    DB       'Hellow! This is test.',ØDH,ØAH
DATA      ENDS

;#########   スタック領域
STACK     SEGMENT STACK
          DW       8ØH DUP (?)
STACK     ENDS
          END      START
```

```
;**********************************************************************
;*                                                                    *
;*      ステータスのチェックをしないでデータを出力する                *
;*                                                                    *
;**********************************************************************
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
START:
        MOV     CX,58                   ;58 文字分繰返し
;                                       ;|
A:      MOV     DL,'A'                  ;"A"を出力
        CALL    PRNT
        LOOP    A                       ;|
;                                       ;LF を出力
        [illegible]
        [illegible]
;                                       ;CR を出力
        [illegible]
        [illegible]
EXIT:                                   ;プログラムを終了
        [illegible]
        [illegible]
PRNT:   [illegible]                     ;センスステータス
        INT     1AH
        [illegible]                     ;受信可,出力
        [illegible]
        [illegible]                     ;BUSY :リトライ
        [illegible]
        [illegible]                     ;ERROR/用紙切れ:終了
        [illegible]
        [illegible]                     ;文字を出力
        [illegible]
        [illegible]
```

```
;**********************************************************************
;*                                                                    *
;*      Ｗａｉｔ時間設定（プリンタの初期化）                          *
;*                                                                    *
;**********************************************************************
CODE    SEGMENT
        ASSUME  CS:CODE,SS:STACK
                                        ;Wait時間設定/初期化
START:  MOV     AH,1BH
        MOV     CX,582
        INT     1AH
;                                       ;プログラムを終了する
        MOV     AH,4CH
        INT     21H

CODE    ENDS

;********  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

# 第8章
# RS-232C BIOS

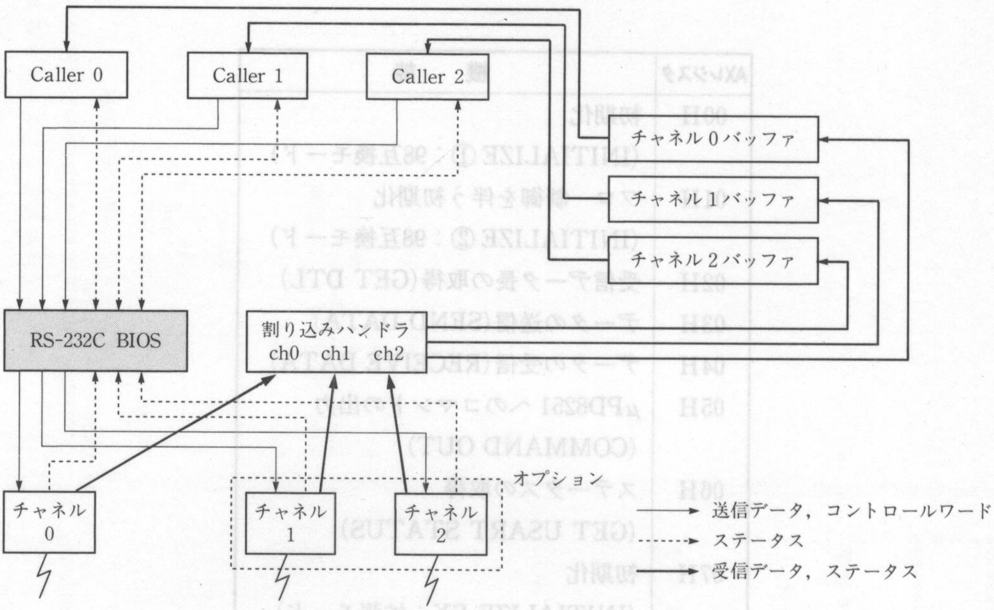
● RS-232C　BIOS 機能一覧

| AXレジスタ | 機　能 |
| --- | --- |
| 00H | 初期化<br>(INITIALIZE ①：98互換モード) |
| 01H | フロー制御を伴う初期化<br>(INITIALIZE ②：98互換モード) |
| 02H | 受信データ長の取得(GET DTL) |
| 03H | データの送信(SEND DATA) |
| 04H | データの受信(RECEIVE DATA) |
| 05H | μPD8251へのコマンドの出力<br>(COMMAND OUT) |
| 06H | ステータスの取得<br>(GET USART STATUS) |
| 07H | 初期化<br>(INITIALIZE EX：拡張モード) |

## 8.1 RS-232C BIOS 共通情報

RS-232C BIOS は，RS-232C インターフェイス規格で接続される周辺機器またはホストコンピュータとの基本的な入出力処理を行う．ただし，本 BIOS では調歩同期式のみをサポートしている．

### (1) データ関連図




### (2) 動作モード

本 BIOS は，PC-9800 シリーズの RS-232C BIOS と互換性のある98互換モード(シングルモード)とマルチタスク対応可能な PC-98XL/XA 固有の拡張モード(マルチモード)を持っている．98互換モードでは，ほとんどの PC-9800 シリーズのソフトウェア資産が活用できるようになっている．いずれのモードも初期化コマンドで設定され，再度初期化されるまで変更できない．

### (3) BIOSコマンド識別コード

本BIOSは，入力のAHレジスタにBIOSコマンド識別コードを指定する.

BIOSコマンド識別コードは，上位4ビットがチャネルの指定(次図参照)，下位4ビットが機能コード(各コマンド参照)となる.

| $b_7$ $b_6$ $b_5$ $b_4$ | RS-232C チャネル指定 |
|---|---|
| 0　0　0　0 | チャネル0 |
| 0　0　0　1 | チャネル1 |
| 0　0　1　0 | チャネル2 |
| その他の指定 | 無効(なにもせずリターンする) |

### (4) システム共通域

| システム共通域<br>シンボル名・絶対番地 | 用　途 |
|---|---|
| RS_CH0_OFST<br>(556H, 557H) | RS-232C受信バッファ領域のオフセットアドレス<br>(チャネル0) |
| RS_CH0_SEG<br>(558H, 559H) | RS-232C受信バッファ領域のセグメントアドレス<br>(チャネル0) |
| RS_CH1_OFST<br>(5F0H, 5F1H) | RS-232C受信バッファ領域のオフセットアドレス<br>(チャネル1) |
| RS_CH1_SEG<br>(5F2H, 5F3H) | RS-232C受信バッファ領域のセグメントアドレス<br>(チャネル1) |
| RS_CH2_OFST<br>(5F4H, 5F5H) | RS-232C受信バッファ領域のオフセットアドレス<br>(チャネル2) |
| RS_CH2_SEG<br>(5F6H, 5F7H) | RS-232C受信バッファ領域のセグメントアドレス<br>(チャネル2) |

注：フィールドはすべて2バイトのワード型

### (5) ハードウェアインターフェイス

①I/Oアドレス

RS-232C BIOSが制御する周辺LSIとそのI/Oアドレス

| LSI | コ　マ　ン　ド | R/W | I/Oアドレス | | |
|---|---|---|---|---|---|
| | | | チャネル0 | チャネル1 | チャネル2 |
| 8251 | Mode/Command | W | 032H | 0B3H | 0BBH |
| | Send/Receive Data | R/W | 030H | 0B1H | 0B9H |
| | Status | R | 032H | 0B3H | 0BBH |
| 8253 | Counter Set | W | 075H | — | — |
| | Counter Mode | W | 077H | — | — |
| 8255 | Mask Set | W | 035H | — | — |
| | Read Signal | R | 033H | — | — |
| 8259 | Read IMR(M) | R | 002H | | |
| | Write OCW1(M) | W | 002H | | |
| | Write OCW2(M) | W | 000H | | |
| | Read IMR(S) | R | 00AH | | |
| | Write OCW1(S) | W | 00AH | | |
| | Write OCW2(S) | W | 008H | | |
| | Write OCW3(S)<br>Read ISR(S) | R/W | 008H | | |
| — | Mask Set | W | — | 0B0H | 0B2H |
| | Read Signal<br>INT Sence | R | — | 0B0H | 0B2H |

② 割り込みレベル

チャネル0の割り込みレベルは，4（マスタ PIC）で固定されている．ベクタ番号は0CH．チャネル1，2ポートの割り込みレベルは，各チャネルで4つの選択が可能である．その内容を次に示す．ただし，各チャネルで同じ割り込みレベルを設定してはいけない．

| | チ　ャ　ネ　ル　1 | チ　ャ　ネ　ル　2 |
|---|---|---|
| 1 | レベル3（マスタ　PIC）<br>ベクタ番号　0BH | レベル3（マスタ　PIC）<br>ベクタ番号　0BH |
| 2 | レベル5（マスタ　PIC）<br>ベクタ番号　0DH | レベル11（スレーブ　PIC）<br>ベクタ番号　13H |
| 3 | レベル6（マスタ　PIC）<br>ベクタ番号　0EH | レベル12（スレーブ　PIC）<br>ベクタ番号　14H |
| 4 | レベル9（スレーブ　PIC）<br>ベクタ番号　10H | レベル13（スレーブ　PIC）<br>ベクタ番号　15H |

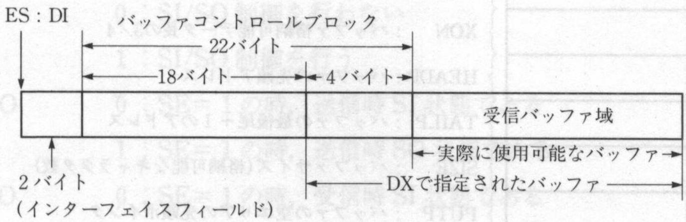
## 8.1.1　受信バッファ

### (1) PC-9800シリーズとの非互換性

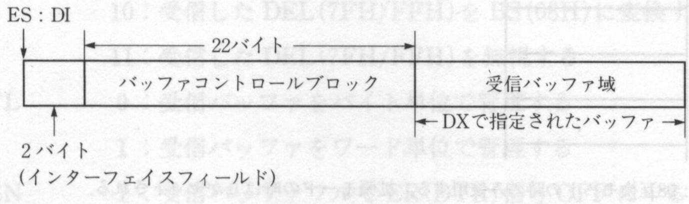
PC-98XL/XAのRS-232C BIOSでは，受信バッファのコントロールブロックとして22バイトの領域を必要とする．

PC-9800シリーズでは，この領域が18バイトであるため，その差4バイトを受信バッファ域を使用することで互換性を保つ．このために初期設定時で示されるバッファ長より実際のバッファ長は4バイト短くなる．98互換モードでは，受信バッファを6バイト以上確保しなければ，バッファとしての役割を果たせない．拡張モードでは，バッファ長はDXで指定した値となる．

●98互換モードでの受信バッファ




●拡張モードでの受信バッファ

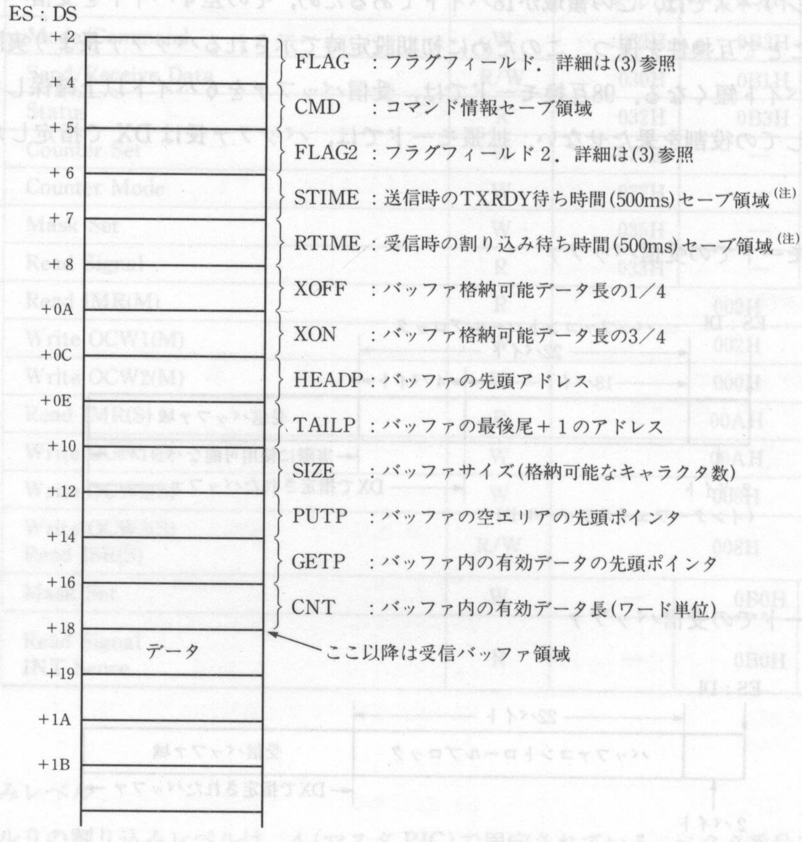



インターフェイスフィールドは2バイトである．ただし，第1バイトの$b_6$～$b_0$と第2バイトは将来用に予約されている．第1バイトの$b_7$(INTフラグ)は，データ受信割り込みが発生しているときに1，発生していないときは0である．ただし，このフラグを参照したら，必ずフラグをリセットする必要がある．

### (2) バッファコントロールブロック

RS-232C BIOSが受信バッファの制御用に使用する領域で，次のようなフィールドに分かれる．

バッファコントロールブロックへの直接アクセスは禁止されている．




注：98互換モードの時のみ使用する．拡張モードの時は0がセットされる．

### (3) フラグフィールド

① 98互換モード

・FLAG

| | | |
|---|---|---|
| $b_{15}$～$b_8$ | 未使用 | 予約 |
| $b_7$ INIT | 1 | RS-232Cポートの初期化済み |
| $b_6$ BFULL | 1 | 受信バッファがフルである |
| $b_5$ OVF | 1 | 受信バッファがオーバーフローした |
| $b_4$ XOFF | 1 | XOFF(CTRL-S出力)を行う |
| $b_3$ XON | 1 | XON(CTRL-Q出力)を行う(XOFF処理済み) |

$b_2$ SXOFF　1：相手局からのXOFF(CTRL-S)受信済み
(XOFFビット＝1のときのみ)
$b_1$，$b_0$　未使用：予約

・FLAG2
$b_7$ MODE　0(常時)：98互換モードで動作中
$b_6$〜$b_0$　未使用：予約

② 拡張モード時

・FLAG
$b_{15}$ INIT　1：RS-232Cポートの初期化済み
$b_{14}$ OVF　1：受信バッファがオーバーフローした
$b_{13}$ SE　0：SI/SO制御を行わない
1：SI/SO制御を行う
$b_{12}$ TXSISO　0：SE＝1の時，送信時SI状態である
1：SE＝1の時，送信時SO状態である
$b_{11}$ RXSISO　0：SE＝1の時，受信時SI状態である
1：SE＝1の時，受信時SO状態である
$b_{10}$〜$b_9$ DELCTL　00：受信したDEL(7FH/FFH)をそのまま渡す
01：受信したDEL(7FH/FFH)をNULL(00H)に変換する
10：受信したDEL(7FH/FFH)をBS(08H)に変換する
11：受信したDEL(7FH/FFH)を無視する
$b_8$ BUFCTL　0：受信バッファをバイト単位で管理する
1：受信バッファをワード単位で管理する
$b_7$ RXEREN　1：受信バッファフルでER(DTR)信号OFFにする
$b_6$ RXEROFF　1：RXEREN＝1の時，ER(DTR)信号OFFである
$b_5$ RXRSEN　1：受信バッファフルでRS(RTS)信号OFFにする
$b_4$ RXRSOFF　1：RXRSEN＝1の時，RS(RTS)信号OFFである
$b_3$ TXE　0：送信時，XON/XOFF処理を行わない
1：送信時，XON/XOFF処理を行う
$b_2$ TXOFF　1：TXE＝1の時，XOFF(DC3)受信済み
$b_1$ RXE　0：受信時，XON/XOFF処理を行わない
1：受信時，XON/XOFF処理を行う
$b_0$RXON　1：RXE＝1の時，XOFF(DC3)送信済み

・FLAG2

$b_7$ MODE　　1（常時）：拡張モードで動作中

$b_6 \sim b_0$　　未使用：予約

### (4) バッファ管理

受信バッファは，初期化コマンドでES：DIで指定された領域の相対18H（98互換モードでは14H）番地からDXで指定されたサイズだけ使用される．

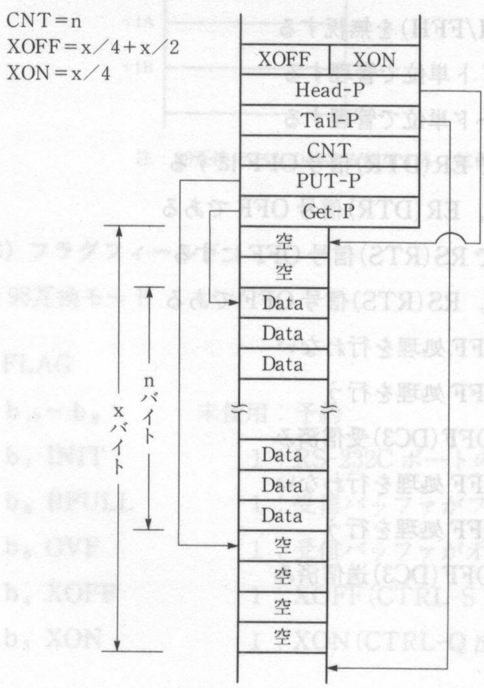
受信バッファはFIFO形式で管理され，その管理情報（ポインタ，カウンタなど）は，バッファコントロールブロック（22バイト：98互換モードでは18バイト）に納められる．

98互換モードの時，受信バッファは受信データ，データ受信時ステータス各1バイト（計2バイト）を1単位として格納する．そのためDXにはバッファに格納可能な受信データ長の2倍＋4（22－18：(1)参照）の値をセットする．

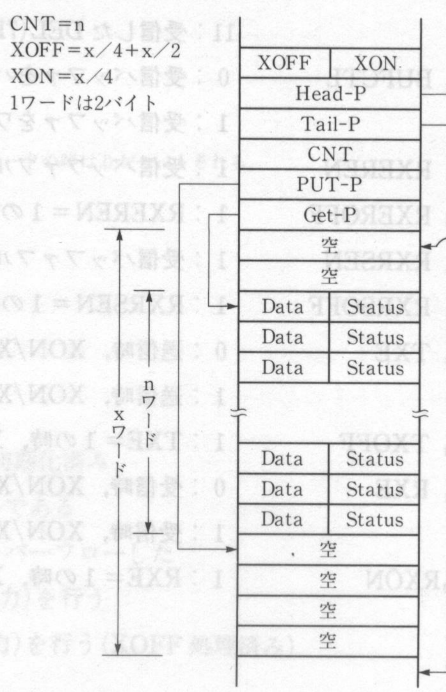
拡張モードの時，初期化コマンド時の指定によって受信バッファへは受信データ，データ受信時ステータス各1バイト（計2バイト）または，受信データ1バイトのみのどちらかを1単位として格納するかを指定できる．そのためDXには受信バッファをバイト（データのみ格納）単位で扱う場合，バッファに格納可能な受信データ長の値をセットする．受信バッファをワード（データとステータス）単位で扱う場合，バッファに格納可能な受信データ長の2倍の値をセットする．

●**拡張モード時バッファ管理図（バイト単位）**

・バッファ管理図（バイト単位）




・バッファ管理図（ワード単位）



### 8.1.2 マルチタスク対応

拡張モード時のRS-232C BIOSは，受信割り込み処理終了時(IRETの直前)に次のファンクションを実行する．

・内部割り込みコード←1FH
・AH ←83H(機能コード)
・AL ←タイプコード
10H：チャネル0
11H：チャネル1
12H：チャネル2

マルチタスク環境をサポートする場合，割り込みコード1FHのベクタを自セグメントに向けて，このRS-232C BIOSからの割り込みを処理する．

AH＝83H，AL＝10H～12H以外の時は元のエントリにJumpさせる必要がある．

なお，このINT 1FHは，1キャラクタ受信毎に発生する．

## 8.2 初期化(INITIALIZE①：98互換モード)

#### (1) 機能

指定されたチャネル番号のRS-232Cポートを初期化する．この時にRS-232C BIOSの動作モードが決定される．初期化する内容は次の通りである．

・μPD8251のモード設定
・タイマμPD8253(ボーレートジェネレータ)の初期設定
・受信バッファ内バッファコントロールブロックの初期設定
・μPD8255への割り込みマスクセット／リセット(チャネル0のみ)
・μPD8259への割り込みマスクセット／リセット
・割り込みベクタの設定(チャネル1，2のみ)

#### (2) 入力

・内部割り込みコード←19H
・AH ←00H(BIOSコマンド識別コード)
・AL ←トランスファレート(チャネル0のみ)

回線上の転送レート(BPS)を指定する．この指定とCHのモードインストラクションによるボーレート指定(×16／×64)によりタイマ(8253)にセットするカウンタを決定する．

チャネル1，2はボード上のディップスイッチで設定する(第2部「11.2 拡張RS-232Cインターフェイス」参照)．

| トランスファレート(AL) | 00H | 01H | 02H | 03H | 04H | 05H | 06H | 07H |
|---|---|---|---|---|---|---|---|---|
| 伝送速度(ボーレート)(bps) | 75 | 150 | 300 | 600 | 1200 | 2400 | 4800 | 9600 |

注：AL に 08H 以上の値を設定した場合は 1200bps とみなされる.

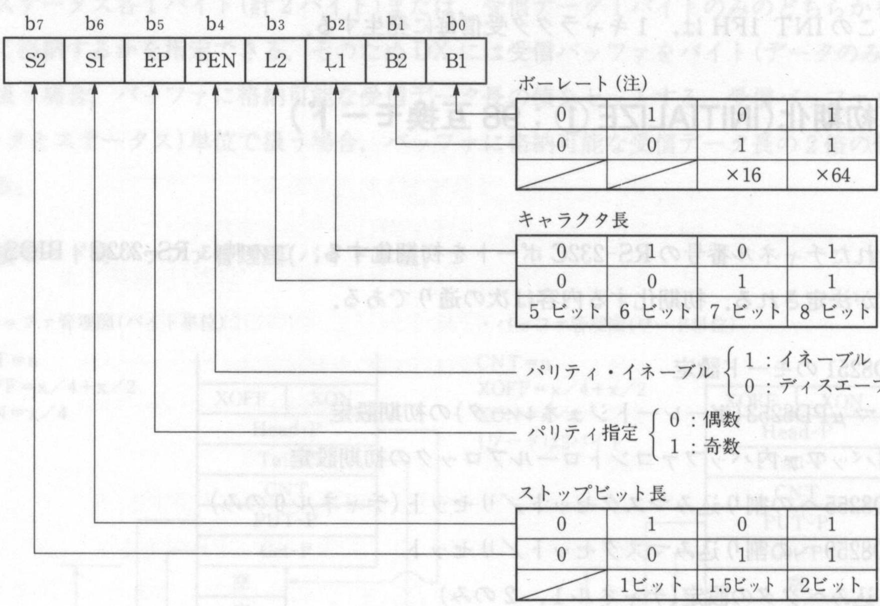
・BH ← 送信時タイムアウト時間

TXRDY ステータス, 送信 XON の待ち時間を 500msec 単位で指定する. 指定範囲は 01H(500msec)～FFH(127.5sec). 00H の時にデフォルト値(02H：1sec)をとる.

・BL ← 受信時タイムアウト時間

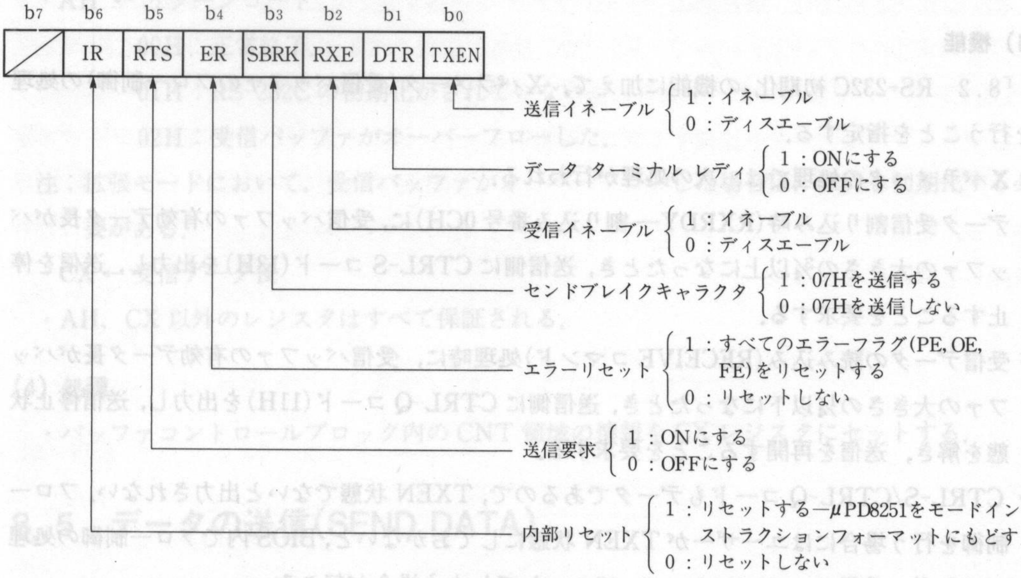
RXRDY 割り込みの待ち時間を500msec単位で指定する. 指定範囲は01H(500msec)～FFH(127.5sec). 00H の時にデフォルト値(3CH：30sec)をとる.

・CH ← μPD8251A モード設定情報(非同期モード)




注：チャネル0のみ有効

・CL　← μPD8251A コマンド指定




・ES/DI ← 受信バッファの先頭アドレス

　受信バッファ形式については８.１.１を参照

・DX　← 受信バッファサイズ(バイト単位)

　受信バッファ形式については８.１.１を参照

**(3) 出力**

・AH　← リターンコード

　00H：正常終了

　04H：拡張 RS-232C インターフェイスボードが未接続

・AH 以外のすべてのレジスタが保証される.

## 8.3　フロー制御を伴う初期化(INITIALIZE②：98互換モード)

### (1) 機能

「8.2　RS-232C 初期化」の機能に加えて，Xパラメータ(受信バッファのフロー制御)の処理を行うことを指定する．

Xパラメータの処理では，次の処理が行われる．

・データ受信割り込み時(RXRDY─割り込み番号0CH)に，受信バッファの有効データ長がバッファの大きさの¾以上になったとき，送信側にCTRL-Sコード(13H)を出力し，送信を停止することを要求する．

・受信データの読み込み(RECEIVEコマンド)処理時に，受信バッファの有効データ長がバッファの大きさの¼以下になったとき，送信側にCTRL-Qコード(11H)を出力し，送信停止状態を解き，送信を再開することを要求する．

・CTRL-S/CTRL-Qコードもデータであるので，TXEN状態でないと出力されない．フロー制御を行う場合にはユーザーがTXEN状態にしておかないと，BIOS内でフロー制御の処理ができず，受信バッファがオーバーフローしてしまう場合が起こる．

### (2) 入力

・内部割り込みコード ← 19H

・AH ← 01H(BIOSコマンド識別コード)

他は「8.2(2)」と同じ．

### (3) 出力条件

「8.2(3)」と同じ．

## 8.4　受信データ長の取得(GET DTL)

### (1) 機能

指定されたチャネル番号の受信バッファ内の有効データ長を得る．

### (2) 入力

・内部割り込みコード ← 19H

・AH ← 02H(BIOSコマンド識別コード)

(3) 出力

・AH ← リターンコード

00H：正常終了.

01H：RS-232C の初期化がされていない.

02H：受信バッファがオーバーフローした.

注：拡張モードにおいて，受信バッファがオーバーフローした場合は，8251を初期化する必要がある.

・CX ← 受信データ長

・AH，CX 以外のレジスタはすべて保証される.

(4) 処理

・バッファコントロールブロック内の CNT 領域の情報を CX レジスタにセットする.

## 8.5 データの送信(SEND DATA)

(1) 機能

指定されたチャネル番号の回線にデータを 1 バイト出力する.

(2) 入力

・内部割り込みコード ← 19H

・AH ← 03H(BIOS コマンド識別コード)

・AL ← 送信データ

・CL ← 参照ステータスビット(拡張モード：(5)参照)

(3) 出力

・AH ← リターンコード

00H：正常終了.

01H：RS-232C インターフェイスの初期設定が行われていない.

02H：受信割り込み処理において，受信バッファがオーバーフローした.

03H：送信，受信処理において，μPD8251 からの送信可，受信可のステータスを引き取れなかった.

05H：送信時 XON/XOFF がイネーブルの時，相手局から XOFF を受信したため，送信停止状態である(SEND DATA のみ).

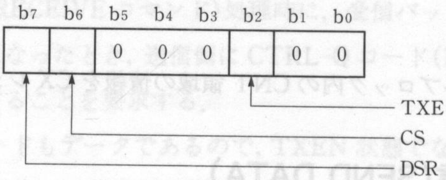
注：拡張モードにおいて，受信バッファがオーバーフローした場合は，8251を初期化する必要がある.

AH 以外のレジスタはすべて保証される.

**(4) 参照するステータス**

RS-232C BIOS では，98互換モード時にデータを送信するときに無条件に参照するステータスはμPD8251 の TXRDY のみで，RS-232C 規格で定義される信号(CS, DTR, CD, DSR, RTS, CI)についてはユーザーが独自に操作，参照しなければならない．

拡張モード時はデータを送信する前に，参照するステータスを表すビット(参照ステータスビット)をCLに指定できる．参照するステータスがTXRDYのみで良い場合は，CLに00Hを指定する．参照ステータスビットの内容を次に示す．他の信号についてはユーザーが独自に操作，参照しなければならない．




**(5) 処理**

① XOFF チェック

送信時Xプロトコル制御イネーブルの時，この XOFF チェックを行う．

XOFF 状態の時，その旨を通知し，異常終了する．

② 送信可チェック

・送信可のステータスをチェックする．

・指定されたステータスがONであることを確認し，その後8251のTXRDYをチェックする．

・指定されたステータスのON，TXRDYが10回のリトライのうちに確認できないときは，その旨を通知し，異常終了する．

③ 送信データのシフト制御

SI/SO 制御イネーブルのとき，キャラクタのシフト制御を行う．

**(6) 送信時Xプロトコル**

RS-232C　BIOSでは PC-9800 シリーズ互換モードでも送信時Xプロトコルを実現する．つまり，互換モードでもホストとしての役割を果たすことが出来る．

送信時Xプロトコル処理は，「フロー制御を伴う初期化」コマンドによって初期化されるとイネーブルとなる．

相手局から XOFF(CTRL-S)を受信すると内部フラグを ON にして(CTRL-S は捨てられる)，送信時に，そのフラグを参照する．

送信時に，その内部フラグがONのとき，指定時間(初期化コマンドで指定されたタイムアウト値)待ち，それでもフラグがOFFにならないときは，その旨を通知する．

## 8.6　データの受信(RECEIVE DATA)

### (1) 機能

指定されたチャネル番号の受信バッファ内の,受信データ(およびステータス)を受け取る(受信バッファが空きでない場合).

受信バッファが空きのとき，タイムアウトのカウンタを設定し，RXRDY の割り込み発生を待つ.

### (2) 入力

・内部割り込みコード ← 19H

・AH ← 04H(BIOS コマンド識別コード)

### (3) 出力

・AH ←リターンコード「8.5(3)」と同じ

ただし，05H は返されない.

・CH ← 受信データ

・CL ← 受信データの受信時ステータス

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| DSR | BD | FE | OE | PE | CI | CS | CD |
| μPD 8251 ステータスバイト | | | | | モデム ステータス | | |

注：各ビットの意味については8.8(3)参照.

・AH, CX 以外のレジスタは保証される.

・拡張モードの初期化コマンドで受信バッファを１バイト単位で管理(キャラクタのみ格納)するように指定した場合は，CL には現在のステータスが返される(形式は受信時ステータスと同じ).

### (4) 処理

① バッファ内有効データチェック

バッファ内に有効データが有れば，データを引き取り，それを返す(このとき，バッファの扱いによりキャラクタのみか，キャラクタとステータスかが決まる). なければ(10回のリトライ後)，異常終了する.

② フラグ制御

正常終了後，内部フラグ(バッファオーバーフロー)をクリアする.

③ 受信時Xプロトコル

データを受信したことによって有効データ長がバッファ長の¼以下になった場合，すでにXOFF(DC3)を送信していることを確かめて相手局にXON(DC1)を送信する．

④ 信号レベルのフロー制御

データを受信したことによってバッファに空きが出来たとき，初期化コマンドで信号レベルのフロー制御が指定されていたら，指定の信号をONにする．

**(5) 受信割り込み処理**

ハードウェアからのRXRDY割り込みによって起動される(チャネル0，1，2共通)．

① 割り込み発生

割り込み発生時，すでに受信バッファがフルだった場合，内部フラグOVF(オーバーフロー)をセットし，処理を終了する．

② データエラーのチェック

PE, FE, OEのうち，いずれかが発生していたら8251にコマンドを出力する．

③ データのセット

8251，8255から受信データ，ステータスを読み取り，ステータスを編集してバッファに格納する(ステータスを格納しないモードもある)．

④ 受信時Xプロトコル

データ長が受信バッファ長の¾以上になったとき，相手局に，XOFF(DC3)コードを送信する．

⑤ 送信時Xプロトコル制御

相手局から送信してくるXOFF，XONコードによって内部フラグを制御する(受信したXOFF, XONコードは捨てられる)．

⑥ キャラクタのシフト制御

受信キャラクタに対してSI/SOシフト制御を行う．SI/SOコードを受信した場合は内部フラグを制御しSI/SOコードは捨てる．

⑦ 信号データのフロー制御

受信データをバッファに格納した後，データ長がバッファ長−1以上になったとき，ER(DTR)またはRS(RTS)あるいは両方の信号をOFFにする(初期化コマンドで指定のあったときのみ)．

⑧ 終了処理

チャネル0は8259に対してEOIを発行する．

チャネル1，2はINT処理に応じたEOI処理(拡張ボードをアクセスしてINTレベルに応じたEOI処理)を行う．

## 8.7　μPD8251へのコマンド出力(COMMAND OUT)

### (1) 機能

指定されたチャネル番号のμPD8251へコマンド情報を出力する.

### (2) 入力

・内部割り込みコード←19H

・AH←05H(BIOSコマンド識別コード)

・AL←μPD8251へ出力するコマンド情報(「8.2(2) CL」参照)

### (3) 出力

・AH ←リターンコード

00H：正常終了

01H：RS-232Cインターフェイスの初期設定が行われていない.

02H：受信バッファがオーバーフローした.

注：拡張モードにおいて，受信バッファがオーバーフローした場合は，8251を初期化する必要がある.

・AH以外のレジスタはすべて保証される.

### (4) 処理

・ALのコマンド情報をμPD8251に出力する.

・コマンド情報でIR(内部リセット)の指示があれば，回線クローズとみなし，バッファコントロールブロック(＋2)FLAGのINITビットをリセットし，RXRDY割り込みをマスクする.

・コマンド情報のRXEビットが0のとき，RXRDY割り込みをマスクする.

## 8.8 ステータスの取得(STATUS)

### (1) 機能

指定されたチャネル番号の μPD8251ステータスとモデムステータスを得る.

### (2) 入力

・内部割り込みコード ← 19H

・AH ← 06H(BIOS コマンド識別コード)

### (3) 出力

・AH ← リターンコード(「8.7(3)」)と同じ.

・CH ← μPD8251 ステータス情報.

| ビット | 略称 | 1 | 0 |
|---|---|---|---|
| $b_7$ | DSR | Date Set Ready ON | Data Set Ready OFF |
| $b_6$ | BD | ブレーク状態検出あり | ブレーク状態検出なし |
| $b_5$ | FE | フレーミングエラー発生 | フレーミングエラーなし |
| $b_4$ | OE | オーバーランエラー発生 | オーバーランエラーなし |
| $b_3$ | PE | パリティエラー発生 | パリティエラーなし |
| $b_2$ | TXE | 送信バッファエンプティ | 送信バッファフル |
| $b_1$ | RXRDY | 受信レディ(受信キャラクタあり) | 受信ビジー |
| $b_0$ | TXRDY | 送信レディ | 送信ビジー |

・CL ← モデムステータス

| ビット | 略称 | 1 | 0 |
|---|---|---|---|
| $b_7$ | $\overline{CI}$ | 着呼なし | 着呼あり |
| $b_6$ | $\overline{CS}$ | 送信不可 | 送信可 |
| $b_5$ | $\overline{CD}$ | 受信キャリア検出なし | 受信キャリア検出 |
| $b_4$ | — | | |
| $b_3$ | — | | |
| $b_2$ | — | | |
| $b_1$ | — | | |
| $b_0$ | — | | |

・AH, CX 以外のレジスタは保証される.

## 8.9　初期化（INITIALIZE EX：拡張モード）

### (1) 機能

指定されたチャネル番号の RS-232C ポートを初期化する．この時にマルチタスク対応の動作モードに設定される．

### (2) 入力

・内部割り込みコード ← 19H

・AH ← 07H（BIOS コマンド識別コード）

・BX ← BIOS 制御情報

| | | | |
|---|---|---|---|
| $b_{15}$ | 未使用 | | ：予約 |
| $b_{14}$ | SISOEN | 0 | ：送受信時に SI/SO 制御を行わない |
| | | 1 | ：送受信時に SI/SO 制御を行う |
| $b_{13}$ | TxXEN | 0 | ：送信側Xプロトコルディスエーブル |
| | | 1 | ：送信側Xプロトコルイネーブル |
| $b_{12}$ | RxXEN | 0 | ：受信側Xプロトコルディスエーブル |
| | | 1 | ：受信側Xプロトコルイネーブル |
| $b_{11}$ | RxEREN | 0 | ：処理しない |
| | | 1 | ：受信バッファフルで ER (DTR)信号 OFF |
| $b_{10}$ | RxRREN | 0 | ：処理しない |
| | | 1 | ：受信バッファフルで RS (RTS)信号 OFF |
| $b_9 \sim b_8$ | DELCTL | 00 | ：受信した DEL (7FH/FFH)をそのまま渡す |
| | | 01 | ：受信した DEL (7FH/FFH)を NULL (00H)に変換 |
| | | 10 | ：受信した DEL (7FH/FFH)を BS (08H)に変換 |
| | | 11 | ：受信した DEL (7FH/FFH)を無視 |
| $b_7 \sim b_1$ | 未使用 | | ：予約 |
| $b_0$ | BUFCTL | 0 | ：受信バッファをバイト単位に管理 |
| | | 1 | ：受信バッファをワード単位に管理 |

バッファ長は DX−1となるので，DX の値はバッファ長の長さ＋1 に設定する．98互換モードで BX に指定するタイムアウトは拡張モードでは存在せず，BIOS 制御情報（BIOS 動作指定）が入る．他は 8．2(2)と同じ

### (3) 出力

8．2(3)と同じ

**●サンプルリスト：** 本サンプルは[5]ページに書かれている環境でテストしたものである．
それ以外の環境での動作は保証できない．

```
;###################################################################
;#                                                                 #
;#      ＲＳ－２３２Ｃを初期化する                                   #
;#                                                                 #
;#   1-STOP BIT / NON PARITY / 8-BITS LENGTH / BAUD RATE 300       #
;#                                                                 #
;###################################################################
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:CODE,ES:BUFFER,SS:STACK
START:
        MOV     AX,CS                   ;データセグメント設定
        MOV     DS,AX
;                                       ;RS-232C初期設定
        MOV     AX,BUFFER               ;バッファセグメント
        MOV     ES,AX
        MOV     DI,OFFSET RS232C        ;バッファオフセット
        MOV     DX,100H                 ;バッファサイズ
        MOV     AH,0                    ;チャネル番号:0
        MOV     AL,2                    ;伝送レート:300BPS
        MOV     CH,01001110B            ;8251モード設定
;                ^^------------------------- Stop bit       :1
;                  ^------------------------ Parity         :-
;                   ^----------------------- Parity   :disable
;                    ^^--------------------- Character Size:8
;                      ^^------------------- Baud Rate  :300bps
        MOV     CL,00110111B            ;8251コマンド設定
;                 ^------------------------- Internal Reset:none
;                  ^------------------------ Request To Send  :ON
;                   ^----------------------- Error Reset    :reset
;                    ^---------------------- Send Break CHR:none
;                     ^--------------------- Recieve Enable:enable
;                      ^-------------------- Data Terminal Ready:ON
;                       ^------------------- Send Enable    :enable
        MOV     BH,10H                  ;送信時Time Out Time
        MOV     BL,10H                  ;受信時Time Out Time
        INT     19H
;                                       ;8251 受信禁止
        MOV     AH,05H
        MOV     AL,00H                  ;8251 コマンド
        INT     19H
;                                       ;プログラムを終了する
        MOV     AH,4CH
        INT     21H
;
CODE    ENDS

;#########  受信バッファ
BUFFER  SEGMENT PARA PUBLIC 'CODE'
RS232C  DB      100H DUP(?)
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;############################################################
;#                                                          #
;#      受信バッファ内の有効データ長を通知する              #
;#                                                          #
;############################################################
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:
        MOV     AX,SEG DATA                 ;データセグメント設定
        MOV     DS,AX
;                                           ;RS-232C初期設定
INIT:   MOV     AX,SEG RS232C                 ;バッファセグメント
        MOV     ES,AX
        MOV     DI,OFFSET RS232C              ;バッファオフセット
        MOV     DX,1ØØH                       ;バッファサイズ
        MOV     AH,Ø                          ;チャネル番号:Ø
        MOV     AL,2                          ;伝送レート:3ØØBPS
        MOV     CH,Ø1ØØ111ØB                  ;8251モード設定
;                ^^-------------------------- Stop bit        :1
;                  ^------------------------- Parity          :-
;                   ^------------------------ Parity   :disable
;                    ^^---------------------- Character Size:8
;                      ^^-------------------- Baud Rate :3ØØbps
        MOV     CL,ØØ11Ø11ØB                  ;8251コマンド設定
;                 ^-------------------------- Internal Reset:none
;                  ^------------------------- Request To Send :ON
;                   ^------------------------ Error Reset  :reset
;                    ^----------------------- Send Break CHR:none
;                     ^---------------------- Recieve Enable:enable
;                      ^--------------------- Data Terminal Ready:ON
;                       ^-------------------- Send Enable  :disable
        MOV     BH,2                          ;送信Time Out Time:2/2sec
        MOV     BL,6Ø                         ;受信Time Out Time:6Ø/2sec
        INT     19H
;                                           ;プロンプトを表示する
        MOV     AH,9
        MOV     DX,OFFSET PRMPT
        INT     21H                         ;キー入力待ち
        MOV     AH,1
        INT     21H
;                                           ;受信バッファのデータ長を得る
GETLEN: MOV     AH,2
        INT     19H
        CMP     AH,Ø                          ;リターンコードをチェック
        JE      NORMAL
;                                           ;エラーメッセージを表示
ERR:    ADD     AH,3ØH
        MOV     ERR_NO,AH
        MOV     AH,9
        MOV     DX,OFFSET ERRMSG
        INT     21H
        JMP     EXIT
;                                           ;データ長を編集
NORMAL: MOV     BX,CX
        MOV     SI,OFFSET LEN
        CALL    CNVNUM
;                                           ;データ長を表示
        MOV     AH,9
        MOV     DX,OFFSET MSG
        INT     21H
EXIT:   MOV     AH,Ø5H                        ;8251 受信禁止
        MOV     AL,ØØH
        INT     19H
;                                           ;プログラムを終了
        MOV     AH,4CH
        INT     21H
CNVNUM: MOV     AX,BX
        XOR     DX,DX
        MOV     CH,5                          ;BINARY ---> NUMERIC
        ADD     SI,5
        MOV     BP,1Ø
CNVLOOP:DIV     BP
        ADD     DL,3ØH
        MOV     DS:[SI],DL
        XOR     DX,DX
        DEC     SI
        DEC     CH
        JNZ     CNVLOOP
CNVRET: RET
CODE    ENDS
;#########    データ領域
DATA    SEGMENT PARA PUBLIC 'CODE'
PRMPT   DB      'データを送信してから何かキーを押して下さい.'
        DB      ØDH,ØAH,'$'
MSG     DB      ØDH,ØAH,'バッファ内のデータ長は,'
LEN     DB      6 DUP(3ØH)
        DB      ' BYTE です.',ØDH,ØAH,'$'
ERRMSG  DB      Ø7H,'エラーが発生しました.',ØDH,ØAH
        DB      'ERROR CODE ='
ERR_NO  DB      ?
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########    受信バッファ
BUFFER  SEGMENT PARA PUBLIC 'CODE'
RS232C  DB      1ØØH DUP(?)
BUFFER  ENDS

;#########    スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;#################################################
;#                                               #
;#      1バイトのデータを送信する                  #
;#                                               #
;#################################################
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメント設定
        MOV     DS,AX
;                                       ;RS-232C初期設定
INIT:   MOV     AX,SEG RS232C           ;バッファセグメント
        MOV     ES,AX
        MOV     DI,OFFSET RS232C        ;バッファオフセット
        MOV     DX,1ØØH                 ;バッファサイズ
        MOV     AH,Ø                    ;チャネル番号:Ø
        MOV     AL,2                    ;伝送レート:3ØØBPS
        MOV     CH,Ø1ØØ111ØB            ;8251モード設定
;                ^^------------------------- Stop bit       :1
;                  ^------------------------ Parity         :-
;                   ^----------------------- Parity  :disable
;                    ^^--------------------- Character Size:8
;                      ^^------------------- Baud Rate :3ØØbps
        MOV     CL,ØØ11ØØØ1B            ;8251コマンド設定
;                ^------------------------- Internal Reset:none
;                 ^------------------------ Send request   :OFF
;                  ^----------------------- Error Reset    :ON
;                   ^---------------------- Send BReak Cha:none
;                    ^--------------------- Recieve Enable:disable
;                     ^-------------------- Data Terminal Ready:OFF
;                      ^------------------- Send Enable    :enable
        MOV     BH,2                    ;送信Time Out Time:2/2sec
        MOV     BL,6Ø                   ;受信Time Out Time:6Ø/2sec
        INT     19H
;                                       ;データを送信する
        MOV     CX,13
        MOV     SI,OFFSET SENDMSG
SEND:   MOV     AL,[SI]
        PUSH    CX
        PUSH    SI
        CALL    SND_CHR                 ;1文字送信
        POP     SI
        POP     CX
        CMP     AH,Ø
        JNE     EXIT
        INC     SI
        LOOP    SEND
;                                       ;メッセージを表示する
        MOV     AH,9
        MOV     DX,OFFSET MSG
        INT     21H
EXIT:
        MOV     AH,Ø5H                  ;8251 受信禁止
        MOV     AL,ØØH
        INT     19H
;
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H
;########  1バイト送信
SND_CHR:MOV     AH,Ø3H                  ;1バイト送信
        MOV     CL,Ø
        INT     19H
        CMP     AH,Ø                    ;リターンコードをチェック
        JE      SND_RET
SND_ERR:PUSH    AX
        ADD     AH,3ØH                  ;リターンコードを文字化
        MOV     ERR_NO,AH               ;リターンコードを表示
        MOV     AH,9
        MOV     DX,OFFSET ERRMSG
        INT     21H
        POP     AX
SND_RET:RET

CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'CODE'
SENDMSG DB      'Test Message',ØDH
MSG     DB      '"Test Message"を送信しました.'
        DB      ØDH,ØAH,'$'
ERRMSG  DB      Ø7H,'エラーが発生しました.',ØDH,ØAH
        DB      'ERROR CODE = '
ERR_NO  DB      ?
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########  受信バッファ
BUFFER  SEGMENT PARA PUBLIC 'CODE'
RS232C  DB      1ØØH DUP(?)
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;#####################################################
;#                                                   #
;#      1バイトのデータを受信する                    #
;#                                                   #
;#####################################################
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:
        MOV     AX,SEG DATA                     ;データセグメント設定
        MOV     DS,AX
                                                ;RS-232C初期設定
INIT:   MOV     AX,SEG RS232C                   ;バッファセグメント
        MOV     ES,AX
        MOV     DI,OFFSET RS232C                ;バッファオフセット
        MOV     DX,100H                         ;バッファサイズ
        MOV     AH,0                            ;チャネル番号:0
        MOV     AL,2                            ;伝送レート:300BPS
        MOV     CH,01001110B                    ;8251モード設定
;                  ^^------------------------- Stop bit         :1
;                    ^------------------------- Parity           :-
;                     ^------------------------ Parity   :disable
;                      ^^---------------------- Character Size:8
;                        ^^-------------------- Baud Rate :300bps
        MOV     CL,00110110B                    ;8251コマンド設定
;                  ^--------------------------- Internal Reset:none
;                   ^-------------------------- Request To Send:ON
;                    ^------------------------- Error Reset :none
;                     ^------------------------ Send BReak Cha:none
;                      ^----------------------- Recieve Enable:enable
;                       ^---------------------- Data Terminal Ready:ON
;                        ^--------------------- Send Enable :disable
        MOV     BH,2                            ;送信Time Out Time:2/2s
        MOV     BL,60                           ;受信Time Out Time:60/2s
        INT     19H
;                                               ;送信を促すメッセージ
        MOV     AH,09H
        MOV     DX,OFFSET MSG
        INT     21H
        MOV     AH,01H                          ;キー入力待ち
        INT     21H
;
RECEIV: MOV     AH,04H                          ;1文字受信
        INT     19H
        CMP     AH,0                            ;リターンコードをチェック
        JNE     ERROR
        CMP     CH,0DH                          ;CR?
        JE      EXIT
;                                               ;受信文字を画面に表示
        MOV     DL,CH
        MOV     AH,02H
        INT     21H
        JMP     RECEIV
;
ERROR:  ADD     AH,30H                          ;リターンコードを文字化
        MOV     ERR_NO,AH
        MOV     AH,9                            ;リターンコードを表示
        MOV     DX,OFFSET ERRMSG
        INT     21H
EXIT:                                           ;8251 受信禁止
        MOV     AH,05H
        MOV     AL,00H
        INT     19H
;                                               ;プログラムを終了
        MOV     AH,4CH
        INT     21H
CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'CODE'
MSG     DB      'データを送信してから何かキーを押して下さい.'
        DB      0DH,0AH,'CR コードの受信で終了します.'
        DB      0DH,0AH,'$'
ERRMSG  DB      07H,'エラーが発生しました.',0DH,0AH
        DB      'ERROR CODE = '
ERR_NO  DB      ?
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  受信バッファ
BUFFER  SEGMENT PARA PUBLIC 'CODE'
RS232C  DB      100H DUP(?)
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      100H DUP(?)
STACK   ENDS

        END     START
```

```
;#####################################################
;#                                                   #
;#      コマンドインストラクションを出力する         #
;#                                                   #
;#      RTS,DTR をオフ．RXE をイネーブルに設定する． #
;#                                                   #
;#####################################################
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメント設定
        MOV     DS,AX
;                                       ;RS-232C初期設定
INIT:   MOV     AX,SEG RS232C            ;バッファセグメント
        MOV     ES,AX
        MOV     DI,OFFSET RS232C         ;バッファオフセット
        MOV     DX,1ØØH                  ;バッファサイズ
        MOV     AH,Ø                     ;チャネル番号:Ø
        MOV     AL,2                     ;伝送レート:3ØØBPS
        MOV     CH,Ø1ØØ111ØB            ;8251モード設定
;                ^^-------------------- Stop bit      :1
;                  ^------------------- Parity        :-
;                   ^------------------ Parity  :disable
;                    ^^---------------- Character Size:8
;                      ^^-------------- Baud Rate :3ØØbps
        MOV     CL,ØØ11ØØ1ØB            ;8251コマンド設定
;                ^--------------------- Internal Reset:none
;                 ^-------------------- Request To Send :ON
;                  ^------------------- Error Reset   :ON
;                   ^------------------ Send BReak Cha:none
;                    ^----------------- Recieve Enable:disable
;                     ^---------------- Data Terminal Ready:ON
;                      ^--------------- Send Enable   :disable
        MOV     BH,2                    ;送信Time Out Time:2/2sec
        MOV     BL,6Ø                   ;受信Time Out Time:6Ø/2sec
        INT     19H
;                                       ;メッセージを表示する
        MOV     AH,9
        MOV     DX,OFFSET MSG
        INT     21H
;                                       ;キー入力待ち
        MOV     AH,Ø1H
        INT     21H
;                                       ;8251コマンドを出力
        MOV     AH,5
        MOV     AL,ØØØ1Ø1Ø1B            ;8251コマンド設定
;                ^--------------------- Internal Reset:none
;                 ^-------------------- Request To Send :OFF
;                  ^------------------- Error Reset   :ON
;                   ^------------------ Send BReak Cha:none
;                    ^----------------- Recieve Enable:enable
;                     ^---------------- Data Terminal Ready:OFF
;                      ^--------------- Send Enable   :enable
        INT     19H
        CMP     AH,Ø                    ;リターンコードチェック
        JE      EXIT
;
ERROR:  ADD     AH,3ØH                  ;リターンコード文字化
        MOV     ERR_NO,AH
        MOV     AH,9                    ;リターンコードを表示
        MOV     DX,OFFSET ERRMSG
        INT     21H
EXIT:                                   ;プログラムを終了
        MOV     AH,4CH
        INT     21H
CODE    ENDS

;#########    データ領域
DATA    SEGMENT PARA PUBLIC 'CODE'
MSG     DB      '何かキーを押すと，コマンドが出力されます．'
        DB      ØDH,ØAH,'$'
ERRMSG  DB      Ø7H,'エラーが発生しました．',ØDH,ØAH
        DB      'ERROR CODE = '
ERR_NO  DB      ?
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########    受信バッファ
BUFFER  SEGMENT PARA PUBLIC 'CODE'
RS232C  DB      1ØØH DUP(?)
BUFFER  ENDS

;#########    スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;#####################################################################
;#                                                                   #
;#      8 2 5 1 のステータスを見る                                   #
;#                                                                   #
;#####################################################################
CODE    SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:
        MOV     AX,SEG DATA             ;データセグメント設定
        MOV     DS,AX
;                                       ;RS-232C初期設定
INIT:   MOV     AX,SEG RS232C              ;バッファセグメント
        MOV     ES,AX
        MOV     DI,OFFSET RS232C           ;バッファオフセット
        MOV     DX,1ØØH                    ;バッファサイズ
        MOV     AH,Ø                       ;チャネル番号:Ø
        MOV     AL,2                       ;伝送レート:3ØØBPS
        MOV     CH,Ø1ØØ111ØB               ;8251モード設定
;                 ^^------------------------ Stop bit         :1
;                   ^----------------------- Parity           :-
;                     ^--------------------- Parity   :disable
;                      ^^------------------- Character Size:8
;                         ^^---------------- Baud Rate :3ØØbps
        MOV     CL,ØØ11Ø11ØB               ;8251コマンド設定
;                 ^------------------------- Internal Reset:none
;                   ^----------------------- Send request   :OFF
;                     ^--------------------- Error Reset    :ON
;                      ^-------------------- Send BReak Cha:none
;                        ^------------------ Recieve Enable:disable
;                         ^----------------- Data Terminal Ready:ON
;                           ^--------------- Send Enable   :enable
        MOV     BH,2                       ;送信Time Out Time:2/2sec
        MOV     BL,6Ø                      ;受信Time Out Time:6Ø/2sec
        INT     19H
;                                       ;ステータス取得
        MOV     AH,6
        INT     19H
        CMP     AH,Ø                    ;リターンコードチェック
        JNE     ERROR
;                                       ;ステータス編集
        XOR     SI,SI
        MOV     AL,CH
        MOV     CX,8
EDIT:   SHL     AL,1
        JB      BIT_1
BIT_Ø:  MOV     AH,'Ø'
        JMP     BIT_MV
BIT_1:  MOV     AH,'1'
BIT_MV: MOV     [STATUS][SI],AH
        INC     SI
        LOOP    EDIT
;                                       ;ステータス表示
        MOV     AH,9
        MOV     DX,OFFSET MSG
        INT     21H
        JMP     EXIT
;
ERROR:  ADD     AH,3ØH                  ;リターンコード文字化
        MOV     ERR_NO,AH
        MOV     AH,9                    ;リターンコードを表示
        MOV     DX,OFFSET ERRMSG
        INT     21H
EXIT:
        MOV     AH,Ø5H                  ;8251 受信禁止
        MOV     AL,ØØH
        INT     19H
;
        MOV     AH,4CH                  ;プログラムを終了
        INT     21H

CODE    ENDS

;#########  データ領域
DATA    SEGMENT PARA PUBLIC 'CODE'
MSG     DB      ØDH,ØAH,'現在の 8251 のステータスは,'
STATUS  DB      8 DUP(?)
        DB      'です.',ØDH,ØAH,'$'
ERRMSG  DB      Ø7H,'エラーが発生しました.',ØDH,ØAH
        DB      'ERROR CODE = '
ERR_NO  DB      ?
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########  受信バッファ
BUFFER  SEGMENT PARA PUBLIC 'CODE'
RS232C  DB      1ØØH DUP(?)
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

# 第9章

# GP-IB BIOS

## ● GP-IB BIOS 機能一覧

| AH レジスタ | 機　　能 |
| --- | --- |
| 00H | 初期化 |
| 01H | IFC の設定 |
| 02H | REN の設定 |
| 03H | REN のリセット |
| 04H | データの送信 |
| 05H | データの受信 |
| 06H | シリアルポールの実行 |
| 07H | SRQ の設定 |
| 08H | パラレルポールの実行 |
| 09H | PPR モードの設定 |
| 0AH | タイムアウトの設定 |
| 0BH | STB のチェック |

## 9.1 GP-IB BIOS 共通情報

### (1) 概要

GP-IB BIOS は PC-9801-29/K GP-IB インターフェイスボード上のハードウェアを制御し, GP-IB に接続される各種周辺機器またはコンピュータとのデータの授受を行うためのものである.

GP-IB BIOS は, 次の機能を提供する.

① 初期設定

GP-IB 用ハードウェアの初期設定および, GP-IB BIOS とユーザー間のインターフェイスの初期設定を行う.

② GP-IB 制御機能

インターフェイスクリア, リモートイネーブル, シリアル／パラレルポール等 GP-IB 接続機器の制御を行う.

③ データ送受信機能

GP-IB 接続機器間でのデータの送受信を行う.

④ GP-IB 監視機能

データ送受信における，時間監視を行う.

### (2) ユーザーインターフェイス

GP-IB BIOS と，ユーザー間の情報の受け渡しは，各種レジスタ，パラメータリストおよびGP-IB 初期設定時に指定される制御情報通知域を通して行う.

### (3) 使用する作業域

GP-IB BIOS では，システム共通域4バイトおよび制御情報通知域の一部を作業域として使用する．また，GP-IB BIOS は割り込み処理ルーチンを含んでおり，割り込み処理中のレジスタ退避等のため，スタックを使用する．GP-IB BIOS で使用するスタックの大きさは，64バイトである.

### (4) GP-IB BIOS の機能

GP-IB BIOS は，複数の GP-IB を制御することは出来ない．また，本ルーチンでサポートする GP-IB 規定の機能サブセットは次のものである.

| 種　別 | 意　　味 |
|---|---|
| SH1 | SH 全機能 |
| AH1 | AH 全機能 |
| T6 | 基礎的トーカ，シリアルポール，MLA によるトーカ解除 |
| TE0 | TE 機能なし |
| L4 | 基本的リスナ，MTA によるリスナ解除 |
| LE0 | LE 機能なし |
| SR1 | SR 全機能 |
| RL1 | RL 全機能 |
| PP1 | リモートメッセージによる構成 |
| DC1 | DC 全機能 |
| DT1 | DT 全機能 |
| C1<br>C2<br>C3<br>C4<br>C26 | システムコントローラ<br>IFC 送信，コントローラ インチャージ<br>REN 送信<br>SRQ に対する応答<br>インターフェイスメッセージ送信，パラレルポール |

(5) 使用方法

GP-IB BIOSの機能を利用するにあたっての，使用方法および制約事項等を次に述べる.

① INTベクタの設定

GP-IB BIOSの各機能の呼び出しは，内部INTコールによるが，ユーザーはこのためのINTベクタをGP-IB BIOS利用前に設定しなければならない.

ユーザーは,適当なINT番号にセグメントアドレスF800H,オフセットアドレス7FDCHを設定しなければならない.

なお, GP-IBからのデータ受信等で使用される割り込みに関しては，ユーザーが関知する必要はない. GP-IB BIOSが, INITIALIZE機能実行時に，ハードウェア上のディップスイッチを参照して，必要なINTベクタを設定する.

② 初期設定

GP-IB BIOS使用にあたっては，各種リソースの初期設定が必要であり，ユーザーは必ずINITIALIZE機能(AH＝00H)を呼び出さなければならない.また,ユーザーがGP-IBコントローラの機能も必要とする場合には，INITIALIZE後，SET IFC機能を呼び出さなければならない(GP-IBコントローラがアクティブな状態となる).

③ 呼び出し方法

GP-IB BIOSを呼び出す時には，パラメータを各レジスタに設定し，①で設定したINT番号をもとに，内部INTでコールする．このとき，SS，SPには，GP-IB BIOSで使用可能なスタックアドレスを設定しておかなければならない.

④ リターン条件

GP-IB BIOSからのリターン時，保証されるレジスタは，DS, SS, SPの3個のレジスタのみである

GP-IB BIOSでの処理結果は，AHレジスタに設定される.

| AH レジスタの内容 | 意　　味 |
|---|---|
| 00H | 処理正常終了<br>RECEIVE DATA 時のデリミタ受信(EOI なし)による終了．<br>EXECUTE SERIAL POLL 時の RQS ビット1の STB 受信による終了(EOI 受信なし)． |
| 01H | RECEIVE DATA 時のデリミタ＋EOI 受信時による終了．<br>EXECUTE SERIAL POLL 時の RQS ビット1かつ EOI 受信ありの STB 受信による終了． |
| 02H | RECEIVE DATA 時の EOI 受信(デリミタなし)による終了． |
| 03H | GP-IB ボードがない(INITIALIZE コマンド時のみ)． |
| 80H | 条件付正常終了．<br>RECEIVE DATA 時の受信データ格納域不足(デリミタ未受信)．<br>EXECUTE SERIAL POLL 時の RQS ビット1の STB 受信なし． |
| FFH | 不正呼び出し |
| FEH | 時間監視による強制終了(タイムアウト)． |
| FCH | エラー検出(GP-IB 規定違反検出)． |
| F8H | IFC 受信による強制終了(スレーブ時)． |

⑤ 注意事項

・GP-IB BIOS 処理中は，外部割り込み可となっている．

・「9.2～9.12」で述べるパラメータの指定値以外の値をパラメータとして指定した場合，結果は保証されない．

⑥ 中断処理ルーチン

GP-IB BIOS では，処理中での中断を可能とするため，INITIALIZE 時に指定されたルーチンに，一定処理ごとに INT 0C5H を行っている．このため，ユーザーは INITIALIZE 時に INT 0C5H に対応するルーチン（IRET のみでも可）のアドレスを設定しておかなければならない．このとき対応するルーチンは，次の制限を守ること．

・セグメントレジスタも含め，全レジスタを保証しなくてはならない．

・ルーチンより，再び GP-IB BIOS を呼び出してはならない．

ただし，GP-IB BIOS の処理続行が不要な場合には，制御を戻す必要はない．

⑦ データ形式

GP-IB BIOS が送受信するデータは，マルチライン上の8ビットのデータである（8ビット ↔ 7ビット等の変換は GP-IB BIOS では行わない）．

⑧ EOS コード

GP-IB BIOS の処理上，$\mu$PD7210（GP-IB コントローラ）に設定する EOS（End of String）コードは，7ビットとして指定する（PC-8097/PC-8897 との互換性のため）．

⑨ 制御情報通知域の再利用

GP-IB BIOS をひとたび INITIALIZE した後，GP-IB の使用を中止する等の理由で，制御情報通知域としていたメモリ領域を他の用途に使用するときは，GP-IB（$\mu$PD7210 からの）

割り込みが発生しないことを保証して(注)から行う必要がある(割り込みが発生すると制御情報通知域であったメモリが書換えられる).

注：μPD8259で割り込みをマスクする等

## (6) 制御情報通知域

| オフセット | | | | |
|---|---|---|---|---|
| 0 | モードおよびMA | Address Status | Interrupt Status 1 | Interrupt Status 2 |
| 4 | GP-IB BIOS作業用(12BYTE) | | | |
| 16 | 上位ルーチンオフセット | | 上位ルーチンセグメント | |
| 20 | | | | |

| 制御情報通知域 | ビット位置 | 意　味 |
|---|---|---|
| モードおよびMA | $b_7$ | 0：拡張INT0使用 |
| | | 1：拡張INT 4／5／6使用 |
| | $b_6$ | 0：IFC未受信 |
| | | 1：IFC受信 |
| | $b_5$(モード) | 0：マスタモード |
| | | 1：スレーブモード |
| | $b_4$～$b_0$ | マイアドレス |
| Address Status | $b_7$(CIC) | 0：コントローラ インアクティブ |
| | | 1：コントローラ アクティブ |
| | $b_6$(ATN) | 未使用(常に0) |
| | $b_5$(SPMS) | 0：シリアルポール実行中ではない |
| | | 1：シリアルポール実行中である |
| | $b_4$(LPAS) | 未使用(常に0) |
| | $b_3$(TPAS) | 未使用(常に0) |
| | $b_2$(LA) | 0：リスナとしてアドレスされていない |
| | | 1：リスナとしてアドレスされている |
| | $b_1$(TA) | 0：トーカとしてアドレスされていない |
| | | 1：トーカとしてアドレスされている |
| | $b_0$(MJMN) | 未使用(常に0) |
| Interrupt Status 1 | $b_7$(CPT) | 未使用(常に0) |
| | $b_6$(APT) | 未使用(常に0) |
| | $b_5$(DET) | 0：デバイストリガ受信なし |
| | | 1：デバイストリガ受信あり |
| | $b_4$(END) | GP-IB BIOSワーク |
| | $b_3$(DEC) | 0：デバイスクリア受信なし |
| | | 1：デバイスクリア受信あり |
| | $b_2$(ERR) | 0：送信正常終了 |
| | | 1：送信異常終了(アクティブなリスナが存在しない.) |
| | $b_1$(DO) | 0：データ送信要求なし |
| | | 1：データ送信要求あり |
| | $b_0$(DI) | 0：データ送信なし |
| | | 1：データ送信あり |

| 制御情報通知域 | ビット位置 | 意　　味 |
|---|---|---|
| Interrupt Status 2 | $b_7$(INT) | 未使用(常に0) |
| | $b_6$(SRQI) | 0：SRQ受信なし<br>1：SRQ受信あり |
| | $b_5$(LOK) | 0：ロックアウト状態でない<br>1：ロックアウト状態である |
| | $b_4$(REM) | 0：リモート状態でない<br>1：リモート状態である |
| | $b_3$(CO) | GP-IB BIOS ワーク |
| | $b_2$(LOCK) | 0：LOKビット変化なし<br>1：LOKビット変化あり |
| | $b_1$(REMC) | 0：REMビット変化なし<br>1：REMビット変化あり |
| | $b_0$(ADSC) | 0：Address StatusのCIC，LA，TAビットの変化なし<br>1：Address StatusのCIC，LA，TAビットのいずれか変化あり |

## (7) システム共通域

| システム共通域<br>シンボル名・絶対番地 | 用　　途 |
|---|---|
| GPIB_WORK<br>(5C2H～5C5H) | GP-IB BIOSの制御情報通知域の先頭アドレス(オフセット，セグメント)を格納する |

## 9.2　初期化(INITIALIZE)

### (1) 機能

GP-IB用ハードウェアの初期設定を行う．また，GP-IB BIOSとユーザー間の情報受渡しの窓口ともなる制御情報通知域を初期設定する．

### (2) 入力

・AH ← 00H(機能コード)

・ES ← 制御情報通知域のセグメントアドレス(0000H～FFFFH)

### (3) 注意事項

① GP-IB BIOS利用に際しては，必ず本機能を呼び出さなければならない．

② 制御情報通知域に設定される情報の種類と形式は前述のとおり．

③ 制御情報通知域の情報は，ERR，DET，DECビット(Interrupt Status 1内)を除いて，未使用も含めすべてGP-IB BIOSが更新する．ユーザーが更新してはならない．

④ Interrupt Status 1内ERR，DET，DECビットは，GP-IB BIOSではオン(1)を設定するだけであり，これらのフラグのオフ(0)はユーザーが行わなければならない(ただし，初期設定時はオフ(0)となっている)．

⑤ モードおよびMA内IFC受信表示ビット($b_6$ビット)は，GP-IB BIOSではオン(1)を設定するだけであり，このフラグのオフ(0)はユーザーが行わなければならない．

⑥ 制御情報通知域は制御情報通知域ベースのオフセット0になければならない．

⑦ アドレスステータス内SPMSビットはコントローラからSTB送出要求が行われた場合に設定され，GP-IB BIOSではオン(1)を設定するだけであり，このフラグのオフ(0)はユーザーが行わなければならない．

## 9.3　IFCの設定(SET IFC)

### (1) 機能

IFCラインを指定の期間アクティブにする．INITIALIZE後，本機能を呼び出すことにより，GP-IBコントローラの機能がアクティブとなる．

### (2) 入力

・AH ← 01H(機能コード)

・BH ← 01H～0FFH

　　指定値×100μSecの間IFCラインをアクティブにする．

**(3) 注意事項**

① 本機能は，GP-IB ハードウェア上のディップスイッチがマスタモードとなっている場合に限り使用可能である．スレーブモードとなっている場合にはエラーリターンする．

② INITIALIZE 後の IFC ラインはインアクティブとなっている．

## 9.4　REN の設定(SET REN)

**(1) 機能**

REN ラインをアクティブにする．

**(2) 入力**

・AH ← 02H(機能コード)

**(3) 注意事項**

① 本機能は，GP-IB コントローラ アクティブの状態でのみ使用可能である．インアクティブ状態の場合はエラーリターンする．

② INITIALIZE 後の REN ラインはインアクティブとなっている．

## 9.5　REN のリセット(RESET REN)

**(1) 機能**

REN ラインをインアクティブにし，約 100μSec 待つ．

**(2) 入力**

・AH ← 03H(機能コード)

**(3) 注意事項**

① 本機能は，GP-IB コントローラ アクティブの状態でのみ使用可能である．インアクティブ状態の場合はエラーリターンする．

② INITIALIZE 後の REN ラインはインアクティブとなっている．

## 9.6 データの送信(SEND DATA)

### (1) 機能

GP-IB上に，コマンドおよびデータを送信する．データ送信の際，指定によっては，データ送信後デリミタ(CR，LF，CR+LF，またはEOI)も送信する．

### (2) 入力

・AH ← 04H(機能コード)
・ES ← コマンドおよびデータ格納域セグメントベース
・SI ← コマンド格納域オフセット
・BX ← コマンド格納域長
・DI ← データ格納域オフセット
・CX ← データ格納域長
・AL ←デリミタ指定

| パラメータ | 指定値 | 意味 |
|---|---|---|
| コマンドおよびデータ格納域 セグメントベース | 0000H ~FFFFH | 送信すべきコマンドおよびデータが格納されている領域のセグメントアドレス |
| コマンド格納域 オフセット | 0000H ~FFFFH | 送信すべきコマンドが格納されている領域のオフセットアドレス |
| コマンド格納域長 | 0000H ~FFFFH | 送信すべきコマンドが格納されている領域の長さ(バイト単位) |
| データ格納域 オフセット | 0000H ~FFFFH | 送信すべきデータが格納されている領域のオフセットアドレス |
| データ格納域長 | 0000H ~FFFFH | 送信すべきデータが格納されている領域の長さ(バイト単位) |
| デリミタ指定 | 00H | デリミタ送信不要 |
| | 01H | デリミタ送信要 デリミタはCR+LF |
| | 02H | デリミタ送信要 デリミタはCR |
| | 03H | デリミタ送信要 デリミタはLF |
| | 80H | デリミタ送信要 デリミタはEOI |
| | 81H | デリミタ送信要 デリミタはCR+LFかつEOI |
| | 82H | デリミタ送信要 デリミタはCRかつEOI |
| | 83H | デリミタ送信要 デリミタはLFかつEOI |

### (3) 注意事項

① コマンドの送信はGP-IBコントローラ アクティブの状態でのみ可能である．インアクティブ状態の場合はエラーリターンする．

② 送信データが無い(データ格納域長＝0)場合，デリミタ指定を80Hとする事は出来ない．指定した場合，デリミタ指定は無視される．

③ 送信するコマンドおよびデータの内容のチェックは行わない．

④ コマンド送信指定がある場合，GP-IBバスは直ちにコマンドモードとなる．なお，データ送受信の完了にかかわらずコマンドモードとなる．

## 9.7 データの受信(RECEIVE DATA)

### (1) 機能

GP-IB上にコマンドを送信後，トーカからのデータを受信する．

### (2) 入力

- ・AH ← 05H(機能コード)
- ・ES ← コマンドおよびデータ格納域セグメントベース
- ・SI ← コマンド格納域オフセット
- ・BX ← コマンド格納域長
- ・DI ← データ格納域オフセット
- ・CX ← データ格納域長
- ・AL ← デリミタ指定

| パラメータ | 指定値 | 意味 |
|---|---|---|
| コマンドおよびデータ格納域 セグメントベース | 0000H ~FFFFH | 送信すべきコマンドおよび受信したデータの格納域セグメントアドレス |
| コマンド格納域 オフセット | 0000H ~FFFFH | 送信すべきコマンドが格納されている領域のオフセットアドレス |
| コマンド格納域長 | 0000H ~FFFFH | 送信すべきコマンドが格納されている領域の長さ(バイト単位) |
| データ格納域 オフセット | 0000H ~FFFFH | 受信したデータが格納される領域のオフセットアドレス |
| データ格納域長 | 0001H ~FFFFH | 受信したデータが格納される領域の長さ(バイト単位) |
| デリミタ指定 | 81H | 受信データ最終はCR+LFまたは，EOI |
| | 82H | 受信データ最終はCRまたはEOI |
| | 83H | 受信データ最終はLFまたはEOI |
| | 80H | 受信データ最終はEOI |

(3) 注意事項

① 本機能の完了は，デリミタ受信時またはデータ格納域へのデータ設定が不可となった時である．どちらの条件で完了したかは，リターンコードにより識別可能である．

② データ設定域へのデータ設定不可による完了の場合，さらにRECEIVE DATA(コマンド送信なし)を呼び出すことにより，後続のデータを受信することが出来る．

③ デリミタは，データ格納域には設定されない．

④ データ格納域に設定された受信データの長さは，DXレジスタにより通知する．デリミタのみ受信時はDX=0となる．

⑤ コマンドの送信はGP-IBコントローラ アクティブの状態でのみ可能である．インアクティブ状態の場合はエラーリターンする．

⑥ データ格納域長を0とすることは出来ない．

⑦ 送信するコマンドの内容のチェックは行わないが，必ずトーカアドレス指定が必要である．

⑧ コマンド送信指定がある場合，GP-IBバスは直ちにコマンドモードとなる．なお，データ送受信の完了にかかわらず，コマンドモードとなる．

## 9.8 シリアルポールの実行(EXECUTE SERIAL POLL)

(1) 機能

指定のトーカに対するシリアルポールを行う．

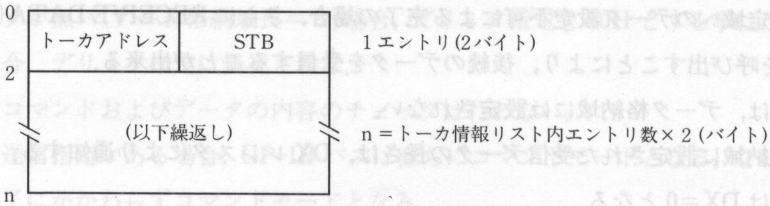
(2) 入力

・AH ← 06H(機能コード)
・ES ← トーカ情報リスト セグメントベース
・DI ← トーカ情報リスト オフセット
・CX ← トーカ情報リスト内エントリ数

| パラメータ | 指定値 | 意味 |
|---|---|---|
| トーカ情報リスト セグメントベース | 0000H ~FFFFH | トーカ情報リストのセグメントアドレス |
| トーカ情報リスト オフセット | 0000H ~FFFFH | トーカ情報リストのオフセットアドレス |
| トーカ情報リスト内 エントリ数 | 0001H ~FFFFH | トーカ情報リスト内のエントリ数 (トーカ数) |

### (3) 注意事項

① トーカ情報リストの形式を次に示す.




② トーカ情報リスト内トーカアドレスは, 呼び出し元で設定する必要があり, 下位5ビットをトーカアドレスとして使用する.

③ トーカ情報リスト内STBはGP-IB BIOSで設定する. STBの設定は, STB内RQSビットが1のトーカアドレスに対応するエントリまでであり, 以降のエントリのSTBはGP-IB BIOSでは設定しない.

④ 本機能の完了は, RQSビットが1のSTB受信時または, 全エントリのSTBを設定した時である. RQSビットが1のSTBを受信したかどうかは, リターンコードにより識別可能である.

⑤ RQSビットが1のSTBを受信した場合, DXレジスタにそのエントリ番号(1～トーカ情報内エントリ数の値)が設定される.

⑥ 時間監視による異常終了(STB受信タイムオーバー)の場合には, DXレジスタにSTB送信を行わなかったトーカアドレスのエントリ番号が設定される.

⑦ 本機能の呼び出しは, GP-IBコントローラ アクティブの状態でのみ可能である. そうでない場合はエラーリターンする.

⑧ 制御情報通知域内SRQIビット(Interrupt Status 2)はRQSビットがオンで, かつSTB受信時のみオフとなる.

⑨ トーカアドレスのチェックは行わない.

⑩ 本機能呼び出しにより, GP-IBバスは直ちにコマンドモードとなる. なお, データ送受信の完了にかかわらずコマンドモードとなる.

## 9.9 SRQの設定(SET SRQ)

### (1) 機能

GP-IB上にSRQを送信し，GP-IBコントローラからのシリアルポールによって，STBを送信する．

### (2) 入力

・AH ←07H(機能コード)
・BH ←STBコード
・BL ←EOI指定

| パラメータ | 指定値 | 意味 |
|---|---|---|
| STBコード | 00H～FFH | シリアルポール時に送信するSTBの値 |
| EOI指定 | 00H | STB送信時EOIを送信しない |
| | 01H | STB送信時EOIを送信する |

### (3) 注意事項

① STBコード中のRQSビット($b_6$ビット)が0の場合，GP-IB上へのSRQの送信は行わない．この場合は，STBの値およびEOI指定を保持するのみである(コントローラからのシリアルポールによって，保持しているSTBが送信される)．

② 初期設定後に保持しているSTBは00H，EOI指定は00Hである．

③ 本機能の呼び出しはGP-IBコントローラ インアクティブの状態でのみ可能である．アクティブの場合はエラーリターンする．

④ 本機能はシリアルポールに対するSTBを保持することで終了する．コントローラからシリアルポールが行われたかどうかは，後述のCHECK STB機能により判別しなければならない．ただし，SRQ送信を行わなかった場合には，CHECK STB機能によりコントローラからシリアルポールが行われたかどうか判別することはできない．

⑤ SRQ送信中，コントローラからのシリアルポール完了以前にRQSビットが0のSTBを指定して本機能を呼び出した場合，SRQの送信は停止する．

## 9.10 パラレルポールの実行(EXECUTE PARALLEL POLL)

### (1) 機能

パラレルポールのライン割り付けおよびパラレルポールの起動，PPRの受信を行う．

### (2) 入力

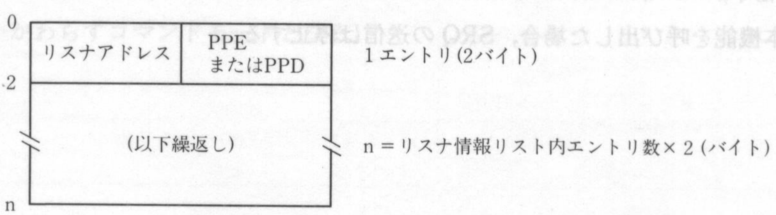
- ・AH ← 08H(機能コード)
- ・ES ← リスナ情報リスト セグメントベース
- ・DI ← リスナ情報リスト オフセット
- ・CX ← リスナ情報リスト内エントリ数
- ・BH ← パラレルポール起動指定
- ・BL ← PPU指定

| パラメータ | 指定値 | 意味 |
|---|---|---|
| リスナ情報リスト セグメントベース | 0000H ~FFFFH | リスナ情報リストのセグメントアドレス |
| リスナ情報リスト オフセット | 0000H ~FFFFH | リスナ情報リストのオフセットアドレス |
| リスナ情報リスト内 エントリ数 | 0000H ~FFFFH | リスナ情報リスト内のエントリ数 (リスナ数) |
| パラレルポール起動指定 | 00H | パラレルポール起動不要 |
| | 01H | パラレルポール起動要 |
| PPU指定 | 00H | パラレルポールのライン割り付け前にPPUを送信しない |
| | 01H | パラレルポールのライン割り付け前にPPUを送信する |

### (3) 注意事項

①パラレルポール起動要を指定した場合，受信したPPRはDHレジスタに設定される．

②リスナ情報リストの形式を次に示す．



③ リスナ情報リストは呼び出し元で設定する必要があり，リスナアドレスは下位5ビットを使用する．

④ 本機能の呼び出しは，GP-IB コントローラ アクティブの状態でのみ可能である．インアクティブの場合はエラーリターンする．

⑤ リスナ情報リスト内エントリ数0，パラレルポール起動要，PPU 送信要の組合せは不可である．ただし，GP-IB BIOS においてチェックは行わない．

⑥ リスナ情報リストの内容のチェックは行わない．

⑦ 本機能呼び出しにより，GP-IB バスは直ちにコマンドモードとなる．なお，データの送受信完了にはかかわらない．

## 9.11 PPR モードの設定(SET PPR MODE)

### (1) 機能

GP-IB コントローラからのパラレルポールに対する応答(PPR)のモードを設定する．

### (2) 入力

・AH ← 09H(機能コード)

・BH ← PPR モード

| パラメータ | 指 定 値 | 意 味 |
|---|---|---|
| PPR モード | 00H | PPR は 0 |
| | 01H | PPR は 1 |
| | 02H | PPR は，SRQ 送信時は 1，SRQ 未送信時は 0 |

### (3) 注意事項

① 初期設定後の PPR モードは 00H である．

② 本機能の呼び出しは GP-IB コントローラ インアクティブの状態でのみ可能である．アクティブの場合はエラーリターンする．

## 9.12　タイムアウトの設定(SET TIME OUT)

### (1) 機能

GP-IBがハングアップしたかどうかを監視するタイムアウトチェックの時間を設定する.

### (2) 入力

・AH ←0AH(機能コード)

・BH ←TIME OUT値

| パラメータ | 指定値 | 意味 |
|---|---|---|
| TIME OUT値 | 00H | タイムアウトチェックを行わない |
| | 01H~FFH | タイムアウトチェック時間(単位：秒) |

### (3) 注意事項

① 初期設定後のTIME OUT値は00H(タイムアウトチェックなし)である.

## 9.13　STBのチェック(CHECK STB)

### (1) 機能

現在保持しているSTBの値，およびEOI指定を通知する.

### (2) 入力

・AH ←0BH(機能コード)

### (3) 注意事項

① 現在保持しているSTBの値をDHレジスタに，EOI指定をDLレジスタに設定する．設定形式は，SET SRQ機能での指定値と同一形式である.

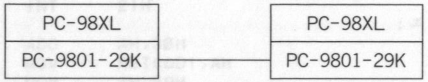
② SET SRQ機能によりSRQの送信を行った後，コントローラからシリアルポールが行われると，本機能で通知するSTB値のRQSビットが0となる．その他のビットおよびEOI指定は，シリアルポールの有無にかかわらずそのまま保持される.

③ 本機能の呼び出しは，GP-IBコントローラ インアクティブの状態でのみ可能である．アクティブの場合はエラーリターンする.

● **サンプルリスト**

本サンプルは次の条件下で動作確認を行った.

・使用機材




・PC-98XLはハイレゾリューションモード，CPUクロック8MHz.

・マスタアドレス＝0，スレーブアドレス＝1.

・MS-DOS………PS98-XL125-HMW

MASM ………PS98-333-HMW

・本サンプルは，各々個別にシステムを立ち上げ直した後に実行すること.

```
;##################################################################
;#                                                                #
;#      ＧＰ－ＩＢボードを初期化する                              #
;#                                                                #
;#        ハードウェア割込み：ＩＲ１２                            #
;#        ＢＩＯＳコマンド  ：ＩＮＴ    ４１Ｈ                    #
;#        中断割込み        ：ＩＮＴ  ０Ｃ５Ｈ                    #
;#                                                                #
;##################################################################

GPIBSEG EQU     ØF8ØØH
GPIBOFF EQU     7FDCH

CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:  MOV     AX,SEG DATA             ;データセグメント設定
        MOV     DS,AX
;                                       ;割込みベクタをセット
        PUSH    DS
        MOV     AX,GPIBSEG
        MOV     DS,AX                           ;GPIB BIOS セグメント
        MOV     DX,GPIBOFF                      ;GPIB BIOS オフセット
        MOV     AH,25H
        MOV     AL,41H                          ;ベクタ番号: 41H
        INT     21H
;
        MOV     AX,INTRPT
        MOV     DS,AX
        MOV     DX,OFFSET INTR
        MOV     AH,25H
        MOV     AL,ØC5H
        INT     21H
        POP     DS
;                                       ;GP_IB イニシャライズ
INIT:   MOV     WORD PTR [BUFFER][16],Ø         ;中断処理ルーチン
        MOV     AX, SEG INTR
        MOV     WORD PTR [BUFFER][18],AX
        MOV     AX,SEG BUFFER
        MOV     ES,AX                           ;制御情報通知域
        MOV     AH,ØØH
        INT     41H
;
        CMP     AH,Ø                            ;エラーチェック
        JNE     ERROR
;                                       ;完了のメッセージを表示
NORMAL: MOV     AH,9
        MOV     DX,OFFSET OKMSG
        INT     21H
;                                       ;721Ø の割込み停止
        CLI
        IN      AL,ØAH
        OR      AH,ØØØ1ØØØØB                    ;IR12(INT 5)の場合
        OUT     ØAH,AL
```

```
;###################################################################
;#                                                                 #
;#         IFCを設定する                                            #
;#                                                                 #
;#         ハードウェア割込み：IR12                                 #
;#         BIOSコマンド     ：INT   41H                             #
;#         中断割込み        ：INT   0C5H                            #
;#                                                                 #
;###################################################################
GPIBSEG EQU     ØF8ØØH
GPIBOFF EQU     7FDCH

CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:  MOV     AX,SEG DATA                 ;データセグメント設定
        MOV     DS,AX
;                                           ;割込みベクタをセット
        PUSH    DS
        MOV     AX,GPIBSEG
        MOV     DS,AX                       ;GPIB BIOS セグメント
        MOV     DX,GPIBOFF                  ;GPIB BIOS オフセット
        MOV     AH,25H
        MOV     AL,41H                      ;ベクタ番号: 41H
        INT     21H
;
        MOV     AX,INTRPT
        MOV     DS,AX
        MOV     DX,OFFSET INTR
        MOV     AH,25H
        MOV     AL,ØC5H
        INT     21H
        POP     DS
;                                           ;GP_IB イニシャライズ
INIT:   MOV     WORD PTR [BUFFER][16],Ø
        MOV     AX, SEG INTR
        MOV     WORD PTR [BUFFER][18],AX    ;中断処理ルーチン
        MOV     AX,SEG BUFFER
        MOV     ES,AX                       ;制御情報通知域
        MOV     AH,ØØH
        INT     41H
;
        CMP     AH,Ø                        ;エラーチェック
        JNE     ERROR1
;                                           ;IFC を設定
IFC:    MOV     AH,1
        MOV     BH,ØAH                      ;アクティブ期間: 1Ø
        INT     41H
        CMP     AH,Ø                        ;エラーチェック
        JNE     ERROR2
;                                           ;721Ø の割込み停止
FINISH: CLI
        IN      AL,ØAH
        STI
;
EXIT:                                       ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;
ERROR:  ADD     AH,3ØH                      ;エラー処理
        MOV     RETCODE,AH                  ;リターンコード表示
        MOV     AH,Ø9H                      ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG
        INT     21H
        JMP     EXIT

CODE    ENDS

;#########   中断処理ルーチン
INTRPT  SEGMENT
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        IRET
INTR    ENDP
INTRPT  ENDS

;#########   データ領域
DATA    SEGMENT
BUFFER  DB      2Ø DUP(?)
OKMSG   DB      'GP_IB ボードを初期化しました.'
        DB      ØDH,ØAH,'$'
ERRMSG  DB      Ø7H
RETCODE DB      ?,' : '
        DB      'GP_IB ボードを初期化できませんでした.'
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########   スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
        OR      AH,ØØØ1ØØØØB                 ;IR12(INT 5)の場合
        OUT     ØAH,AL
        STI
;
EXIT:                                        ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;                                            ;エラー処理(INITIALIZE)
ERROR1: ADD     AH,3ØH                        ;リターンコード
        MOV     RETCOD1,AH
        MOV     AH,Ø9H                        ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG1
        INT     21H
        JMP     EXIT
;                                            ;エラー処理(IFC)
ERROR2: ADD     AH,3ØH                        ;リターンコード
        MOV     RETCOD2,AH
        MOV     AH,Ø9H                        ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG2
        INT     21H
        JMP     FINISH

CODE    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        IRET
INTR    ENDP
INTRPT  ENDS

;#########  データ領域
DATA    SEGMENT
BUFFER  DB      2Ø DUP(?)
ERRMSG1 DB      Ø7H
RETCOD1 DB      ?,' : '
        DB      'GP_IB ボードを初期化できませんでした.'
        DB      ØDH,ØAH,'$'
ERRMSG2 DB      Ø7H
RETCOD2 DB      ?,' : '
        DB      'GP_IB BIOS エラーリターン.'
        DB      ØDH,ØAH,'$'
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      8ØH DUP (?)
STACK   ENDS

        END     START
```

```
;##################################################################
;#                                                                #
;#      RENを設定する                                             #
;#                                                                #
;#      ハードウェア割込み : IR12                                 #
;#      BIOSコマンド       : INT   41H                            #
;#      中断割込み         : INT  0C5H                            #
;#                                                                #
;##################################################################

GPIBSEG EQU     ØF8ØØH
GPIBOFF EQU     7FDCH

CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:  MOV     AX,SEG DATA                  ;データセグメント設定
        MOV     DS,AX
;                                            ;割込みベクタをセット
        PUSH    DS
        MOV     AX,GPIBSEG
        MOV     DS,AX                         ;GPIB BIOS セグメント
        MOV     DX,GPIBOFF                    ;GPIB BIOS オフセット
        MOV     AH,25H
        MOV     AL,41H                        ;ベクタ番号: 41H
        INT     21H
;
        MOV     AX,INTRPT
        MOV     DS,AX
        MOV     DX,OFFSET INTR
        MOV     AH,25H
        MOV     AL,ØC5H
        INT     21H
        POP     DS
;                                            ;GP_IB イニシャライズ
INIT:   MOV     WORD PTR [BUFFER][16],Ø
        MOV     AX, SEG INTR
        MOV     WORD PTR [BUFFER][18],AX      ;中断処理ルーチン
        MOV     AX,SEG BUFFER
        MOV     ES,AX                         ;制御情報通知域
        MOV     AH,ØØH
        INT     41H
        CMP     AH,Ø                          ;エラーチェック
        JNE     ERROR1
;                                            ;IFC を設定
IFC:    MOV     AH,Ø1H
        MOV     BH,ØAH                        ;アクティブ期間: 1Ø
        INT     41H
        CMP     AH,Ø                          ;エラーチェック
        JNE     ERROR2
```

```
;                                       ;REN を設定
REMOTE: MOV     AH,02H
        INT     41H
        CMP     AH,0                      ;エラーチェック
        JNE     ERROR2
;                                       ;7210 の割込み停止
FINISH: CLI
        IN      AL,0AH
        OR      AH,00010000B            ;IR12(INT 5)の場合
        OUT     0AH,AL
        STI
;
EXIT:                                   ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;                                       ;エラー処理(INITIALIZE)
ERROR1: ADD     AH,30H                    ;リターンコード
        MOV     RETCOD1,AH
        MOV     AH,09H                    ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG1
        INT     21H
        JMP     EXIT
;                                       ;エラー処理(その他)
ERROR2: ADD     AH,30H                    ;リターンコード
        MOV     RETCOD2,AH
        MOV     AH,09H                    ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG2
        INT     21H
        JMP     FINISH

CODE    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
          IRET
INTR    ENDP
INTRPT  ENDS

;#########  データ領域
DATA    SEGMENT
BUFFER  DB      20 DUP(?)
ERRMSG1 DB      07H
RETCOD1 DB      ?,' : '
        DB      'GP_IB ボードを初期化できませんでした.'
        DB      0DH,0AH,'$'
ERRMSG2 DB      07H
RETCOD2 DB      ?,' : '
        DB      'GP_IB BIOS エラーリターン.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;################################################################
;#                                                              #
;#      R E Nをリセットする                                      #
;#                                                              #
;#      ハードウェア割込み : I R 1 2                             #
;#      B I O Sコマンド    : I N T    4 1 H                      #
;#      中断割込み         : I N T  0 C 5 H                      #
;#                                                              #
;################################################################

GPIBSEG EQU     ØF8ØØH
GPIBOFF EQU     7FDCH

CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,SS:STACK
START:  MOV     AX,SEG DATA                 ;データセグメント設定
        MOV     DS,AX
;                                           ;割込みベクタをセット
        PUSH    DS
        MOV     AX,GPIBSEG
        MOV     DS,AX                       ;GPIB BIOS セグメント
        MOV     DX,GPIBOFF                  ;GPIB BIOS オフセット
        MOV     AH,25H
        MOV     AL,41H                      ;ベクタ番号: 41H
        INT     21H
;
        MOV     AX,INTRPT
        MOV     DS,AX
        MOV     DX,OFFSET INTR
        MOV     AH,25H
        MOV     AL,ØC5H
        INT     21H
        POP     DS
;                                           ;GP_IB イニシャライズ
INIT:   MOV     WORD PTR [BUFFER][16],Ø
        MOV     AX, SEG INTR
        MOV     WORD PTR [BUFFER][18],AX    ;中断処理ルーチン
        MOV     AX,SEG BUFFER
        MOV     ES,AX                       ;制御情報通知域
        MOV     AH,ØØH
        INT     41H
        CMP     AH,Ø                        ;エラーチェック
        JNE     ERROR1
;                                           ;IFC を設定
IFC:    MOV     AH,Ø1H
        MOV     BH,ØAH                      ;アクティブ期間: 1Ø
        INT     41H
        CMP     AH,Ø                        ;エラーチェック
        JNE     ERROR2
;                                           ;REN を設定
REMOTE: MOV     AH,Ø2H
        INT     41H
        CMP     AH,Ø                        ;エラーチェック
        JNE     ERROR2
;                                           ;REN をリセット
RESET:  MOV     AH,Ø3H
        INT     41H
        CMP     AH,Ø                        ;エラーチェック
        JNE     ERROR2
;                                           ;721Ø の割込み停止
FINISH: CLI
        IN      AL,ØAH
        OR      AH,ØØØ1ØØØØB                ;IR12(INT 5)の場合
        OUT     ØAH,AL
        STI
;                                           ;プログラムを終了
EXIT:
        MOV     AH,4CH
        INT     21H
;                                           ;エラー処理(INITIALIZE)
ERROR1: ADD     AH,3ØH                      ;リターンコード
        MOV     RETCOD1,AH
        MOV     AH,Ø9H                      ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG1
        INT     21H
        JMP     EXIT
;                                           ;エラー処理(その他)
ERROR2: ADD     AH,3ØH                      ;リターンコード
        MOV     RETCOD2,AH
        MOV     AH,Ø9H                      ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG2
        INT     21H
        JMP     FINISH

CODE    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        IRET
INTR    ENDP
INTRPT  ENDS

;#########  データ領域
DATA    SEGMENT
BUFFER  DB      2Ø DUP(?)
ERRMSG1 DB      Ø7H
RETCOD1 DB      ?,' : '
        DB      'GP_IB ボードを初期化できませんでした.'
        DB      ØDH,ØAH,'$'
ERRMSG2 DB      Ø7H
RETCOD2 DB      ?,' : '
        DB      'GP_IB BIOS エラーリターン.'
        DB      ØDH,ØAH,'$'
DATA    ENDS
```

```
;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;###############################################################
;#                                                             #
;#      データを送信する                                        #
;#                                                             #
;#         ハードウェア割込み : IR12                           #
;#         BIOSコマンド       : INT    41H                     #
;#         中断割込み         : INT   0C5H                     #
;#                                                             #
;###############################################################
GPIBSEG EQU     0F800H
GPIBOFF EQU     7FDCH

CODE    SEGMENT
        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:  MOV     AX,SEG DATA                     ;データセグメント設定
        MOV     DS,AX
;                                               ;割込みベクタをセット
        PUSH    DS
        MOV     AX,GPIBSEG
        MOV     DS,AX                             ;GPIB BIOS セグメント
        MOV     DX,GPIBOFF                        ;GPIB BIOS オフセット
        MOV     AH,25H
        MOV     AL,41H                            ;ベクタ番号: 41H
        INT     21H
;
        MOV     AX,INTRPT
        MOV     DS,AX
        MOV     DX,OFFSET INTR
        MOV     AH,25H
        MOV     AL,0C5H
        INT     21H
        POP     DS
;                                               ;GP_IB イニシャライズ
INIT:   MOV     AX,SEG BUFFER                     ;制御情報通知域
        MOV     ES,AX
        MOV     WORD PTR ES:[16],0              ;中断処理オフセット
        MOV     AX, SEG INTR
        MOV     WORD PTR ES:[18],AX             ;中断処理セグメント
        MOV     AH,00H
        INT     41H
        CMP     AH,0                              ;エラーチェック
        JNE     ERROR1
;                                               ;IFC を設定
IFC:    MOV     AH,01H
        MOV     BH,0AH                            ;アクティブ期間: 10
        INT     41H
        CMP     AH,0                              ;エラーチェック
        JNE     ERROR2
;                                               ;REN を設定
REMOTE: MOV     AH,02H
        INT     41H
        CMP     AH,0                              ;エラーチェック
```

```
        JNE     ERROR2
;                                       ;データを送信
        MOV     AX,SEG BUFFER               ;制御情報通知域
        MOV     ES,AX
        AND     BYTE PTR ES:[0],10111111B   ;IFC 未受信 OFF
        AND     BYTE PTR ES:[1],11011111B   ;SPMS OFF
        AND     BYTE PTR ES:[2],11010011B   ;DET/DEC/ERR OFF
        MOV     SI,OFFSET COMM              ;コマンド域オフセット
        MOV     BX,3                        ;コマンド域長
        MOV     DI,OFFSET DATUM             ;データ域オフセット
        MOV     CX,18                       ;データ域長
        MOV     AH,4
        MOV     AL,81H                      ;デリミタ送信要(EOI)
        INT     41H
        CMP     AH,0                        ;エラーチェック
        JNE     ERROR2
;                                       ;7210 の割込み停止
FINISH: CLI
        IN      AL,0AH
        OR      AH,00010000B                ;IR12(INT 5)の場合
        OUT     0AH,AL
        STI
;
EXIT:                                   ;プログラムを終了
        MOV     AH,4CH
        INT     21H
;                                       ;エラー処理(INITIALIZE)
ERROR1: ADD     AH,30H                      ;リターンコード
        MOV     RETCOD1,AH
        MOV     AH,09H                      ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG1
        INT     21H
        JMP     EXIT
;                                       ;エラー処理(その他)
ERROR2: ADD     AH,30H                      ;リターンコード
        MOV     RETCOD2,AH
        MOV     AH,09H                      ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG2
        INT     21H
        JMP     FINISH

CODE    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        IRET
INTR    ENDP
INTRPT  ENDS

;#########  データ領域
DATA    SEGMENT
ERRMSG1 DB      07H
RETCOD1 DB      ?,' : '
        DB      'GP_IB ボードを初期化できませんでした.'
        DB      0DH,0AH,'$'
ERRMSG2 DB      07H
RETCOD2 DB      ?,' : '
        DB      'GP_IB BIOS エラーリターン.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  GP_IB BIOS バッファ
BUFFER  SEGMENT
BUFF    DB      20 DUP(?)
COMM    DB      3FH
        DB      40H
        DB      21H
DATUM   DB      'SEND-DATA-FROM-MST'
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
;#################################################################
;#                                                               #
;#    データを受信する                                             #
;#                                                               #
;#       ハードウェア割込み : IR12                                 #
;#       BIOSコマンド     : INT    41H                            #
;#       中断割込み       : INT   0C5H                            #
;#                                                               #
;#################################################################

GPIBSEG EQU      ØF8ØØH
GPIBOFF EQU      7FDCH

CODE     SEGMENT
         ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
START:   MOV     AX,SEG DATA                 ;データセグメント設定
         MOV     DS,AX
;                                            ;割込みベクタをセット
         PUSH    DS
         MOV     AX,GPIBSEG
         MOV     DS,AX                       ;GPIB BIOS セグメント
         MOV     DX,GPIBOFF                  ;GPIB BIOS オフセット
         MOV     AH,25H
         MOV     AL,41H                      ;ベクタ番号: 41H
         INT     21H
;
         MOV     AX,INTRPT
         MOV     DS,AX
         MOV     DX,OFFSET INTR
         MOV     AH,25H
         MOV     AL,ØC5H
         INT     21H
         POP     DS
;                                            ;GP_IB イニシャライズ
INIT:    MOV     AX,SEG BUFFER               ;制御情報通知域
         MOV     ES,AX
         MOV     WORD PTR ES:[16],Ø          ;中断処理オフセット
         MOV     AX, SEG INTR
         MOV     WORD PTR ES:[18],AX         ;中断処理セグメント
         MOV     AH,ØØH
         INT     41H
         CMP     AH,Ø                        ;エラーチェック
         JNE     ERROR1
;                                            ;IFC を設定
IFC:     MOV     AH,Ø1H
         MOV     BH,1Ø                       ;アクティブ期間: 1Ø
         INT     41H
         CMP     AH,Ø                        ;エラーチェック
         JE      REMOTE
         JMP     ERROR2
;                                            ;REN を設定
REMOTE:  MOV     AH,Ø2H
         INT     41H
         CMP     AH,Ø                        ;エラーチェック
         JNE     ERROR2
;                                            ;タイムアウトの設定
T_OUT:   MOV     AX,SEG BUFFER               ;制御情報通知域
         MOV     ES,AX
         MOV     AH,ØAH
         MOV     BH,5                        ;タイムアウト値: 5
         INT     41H
;                                            ;データを受信
RECEIV:  MOV     AX, SEG BUFFER              ;制御情報通知域
         MOV     ES,AX
         AND     BYTE PTR ES:[Ø],1Ø111111B   ;IFC 未受信 OFF
         AND     BYTE PTR ES:[1],11Ø11111B   ;SPMS OFF
         AND     BYTE PTR ES:[2],11Ø1ØØ11B   ;DET/DEC/ERR OFF
         MOV     SI,OFFSET COMMAND           ;コマンド域オフセット
         MOV     BX,3                        ;コマンド域長
         MOV     DI,OFFSET DATUM             ;データ域オフセット
         MOV     CX,1ØØH                     ;受信バッファ長
         MOV     AH,Ø5H
         MOV     AL,81H                      ;デリミタ送信要(EOI)
         INT     41H
         CMP     AH,Ø                        ;エラーチェック
         JNE     ERROR2
;                                            ;受信データを表示する
         MOV     AH,Ø9H                      ;"受信データは"
         MOV     DX,OFFSET MSG
         INT     21H
         XOR     SI,SI                       ;"x..."
         MOV     BX,OFFSET ES:[DATUM]
;
DISP:    MOV     DL,ES:[BX][SI]
         CMP     DL,Ø
         JE      EOI
         MOV     AH,Ø2H
         INT     21H
         INC     SI
         JMP     DISP
;
EOI:     MOV     AH,Ø9H                      ;"です"
         MOV     DX,OFFSET DESU
         INT     21H
;                                            ;721Ø の割込み停止
FINISH:  CLI
         IN      AL,ØAH
         OR      AH,ØØØ1ØØØØB                ;IR12(INT 5)の場合
         OUT     ØAH,AL
         STI
;
EXIT:                                        ;プログラムを終了
         MOV     AH,4CH
         INT     21H
;                                            ;エラー処理(INITIALIZE)
ERROR1:  ADD     AH,3ØH                      ;リターンコード
```

```
        MOV     RETCOD1,AH
        MOV     AH,09H                          ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG1
        INT     21H
        JMP     EXIT
;                                           ;エラー処理(その他)
ERROR2: ADD     AH,30H                          ;リターンコード
        MOV     RETCOD2,AH
        MOV     AH,09H                          ;エラーメッセージを表示
        MOV     DX,OFFSET ERRMSG2
        INT     21H
        JMP     FINISH

CODE    ENDS

;#########  中断処理ルーチン
INTRPT  SEGMENT
        ASSUME  CS:INTRPT,DS:INTRPT
INTR    PROC    FAR
        IRET
INTR    ENDP
INTRPT  ENDS

;#########  データ領域
DATA    SEGMENT
MSG     DB      0DH,0AH,'受信したデータは'
        DB      0DH,0AH,'$'
DESU    DB      0DH,0AH,'です.$'
ERRMSG1 DB      07H
RETCOD1 DB      ?,' : '
        DB      'GP_IB ボードを初期化できませんでした.'
        DB      0DH,0AH,'$'
ERRMSG2 DB      07H
RETCOD2 DB      ?,' : '
        DB      'GP_IB BIOS エラーリターン.'
        DB      0DH,0AH,'$'
DATA    ENDS

;#########  GP_IB BIOS バッファ
BUFFER  SEGMENT
BUFF    DB      20 DUP(?)
COMMAND DB      3FH
        DB      41H
        DB      20H
DATUM   DB      100H dup(?)
BUFFER  ENDS

;#########  スタック領域
STACK   SEGMENT STACK
        DW      80H DUP (?)
STACK   ENDS

        END     START
```

```
*            システム割込み : IR12                *
*            BIOSコール     : INT 41H             *
*            [illegible]    : INT 0C5H            *

        ASSUME  CS:CODE,DS:DATA,ES:BUFFER,SS:STACK
[illegible] MOV AX,SEG DATA                 ;データセグメント設定
        MOV     DS,AX
                                   ;GPIB BIOS 割込みベクタセット
        MOV     [illegible]
        MOV     [illegible]                 ;GPIB BIOS セグメント
        MOV     DX,OFFSET [illegible]       ;GPIB BIOS オフセット
        MOV     AX,25[illegible]
        MOV     AL,41H                      ;ベクタ番号：41H
        [illegible]
        MOV     DS,AX
        MOV     DX,OFFSET INTR
        [illegible]
        INT     21H
        POP     DS
                                   ;GP-IB イニシャライズ
INIT    [illegible]                ;[illegible]
        MOV     ES,AX
        MOV     WORD PTR ES:[16],0          ;中断処理オフセット
        MOV     AX,SEG INTR
        MOV     WORD PTR ES:[18],AX         ;中断処理セグメント
        MOV     AH,00H
        [illegible]
        INT     41H
        CMP     AH,0                        ;エラーチェック
        [illegible]
                                   ;IFC を送出 [illegible]
IFC:    MOV     AH,[illegible]
        MOV     [illegible]
        MOV     AH,0[illegible]             ;[illegible]
        INT     41H
        CMP     AH,0                        ;エラー処理(その他)
        JNE     ERROR[illegible]
                                   ;REN を設定 [illegible]
REMOTE: MOV     AH,[illegible]
        INT     41H
        CMP     AH,0                        ;エラーチェック
        JNE     ERROR2
                                   ;タイムアウトの設定
T_OUT   MOV     AX,SEG BUFFER               ;[illegible]
        MOV     ES,AX
        MOV     AH,0AH
        MOV     BH,5                        ;タイムアウト値：5
        INT     41H
                                   ;データを受信
RECEIV: MOV     AX,SEG BUFFER               ;[illegible]
        MOV     ES,AX
        AND     BYTE PTR ES:[0],10111111B   ;IFC 未受信 OFF
        AND     BYTE PTR ES:[1],11011111B   ;SRQS OFF
        AND     BYTE PTR ES:[2],11000011B   ;DET/DEC/ERR OFF
        MOV     SI,OFFSET COMMAND           ;コマンド域オフセット
        MOV     BX,3                        ;コマンド域長
        MOV     DI,OFFSET DATUM             ;データ域オフセット
        MOV     CX,100H                     ;受信バッファ長
        MOV     AH,05H
        MOV     AL,0[illegible]             ;デリミタ送信要(EOI)
        INT     41H
        CMP     AH,0                        ;エラーチェック
        JNE     ERROR2
                                   ;受信データを表示する
        MOV     AH,09H                      ;"受信データは"
        MOV     DX,OFFSET MSG
        INT     21H
        XOR     SI,SI                       ;"x  "
        MOV     BX,OFFSET ES:[DATUM]

DISP:   MOV     DL,ES:[BX][SI]
        CMP     DL,0
        JE      EOI
        MOV     AH,02H
        INT     21H
        INC     SI
        JMP     DISP

EOI:    MOV     AH,09H                      ;"です。"
        MOV     DX,OFFSET DESU
        INT     21H
                                   ;IR12 の割込み停止
FINISH: CLI
        IN      AL,0AH
        OR      AH,00010000B                ;IR12(INT 5)の場合
        OUT     0AH,AL
        STI
                                   ;プログラムを終了
EXIT:
        MOV     AH,4CH
        INT     21H
                                   ;エラー処理(INITIALIZE)
ERROR1: ADD     AH,30H                      ;リターンコード
```

# 第4部
# 外部インターフェイス仕様

# 第1章

# 拡張スロットバスインターフェイス

## 1.1　外部仕様

### (1) 物理的なバスコネクタの信号名と形状

①バススロット信号

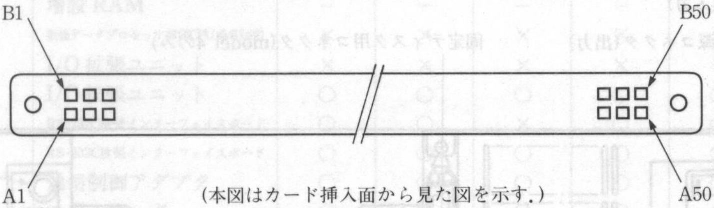
| 端子番号 | 信号名 | 方向 | 機能 | 端子番号 | 信号名 | 方向 | 機能 |
|---|---|---|---|---|---|---|---|
| A 1 | GND | | | B 1 | GND | | |
| A 2 | V1 | | | B 2 | V1 | | |
| A 3 | V2 | | | B 3 | V2 | | |
| A 4 | AB001 | I／O | アドレスバス | B 4 | DB001 | I／O | データバス |
| A 5 | AB011 | 〃 | 〃 | B 5 | DB011 | 〃 | 〃 |
| A 6 | AB021 | 〃 | 〃 | B 6 | DB021 | 〃 | 〃 |
| A 7 | AB031 | 〃 | 〃 | B 7 | DB031 | 〃 | 〃 |
| A 8 | AB041 | 〃 | 〃 | B 8 | DB041 | 〃 | 〃 |
| A 9 | AB051 | 〃 | 〃 | B 9 | DB051 | 〃 | 〃 |
| A10 | AB061 | 〃 | 〃 | B10 | DB061 | 〃 | 〃 |
| A11 | GND | | | B11 | GND | | |
| A12 | AB071 | I／O | アドレスバス | B12 | DB071 | I／O | データバス |
| A13 | AB081 | 〃 | 〃 | B13 | DB081 | 〃 | 〃 |
| A14 | AB091 | 〃 | 〃 | B14 | DB091 | 〃 | 〃 |
| A15 | AB101 | 〃 | 〃 | B15 | DB101 | 〃 | 〃 |
| A16 | AB111 | 〃 | 〃 | B16 | DB111 | 〃 | 〃 |
| A17 | AB121 | 〃 | 〃 | B17 | DB121 | 〃 | 〃 |
| A18 | AB131 | 〃 | 〃 | B18 | DB131 | 〃 | 〃 |
| A19 | AB141 | 〃 | 〃 | B19 | DB141 | 〃 | 〃 |
| A20 | AB151 | 〃 | 〃 | B20 | DB151 | 〃 | 〃 |
| A21 | GND | | | B21 | GND | | |
| A22 | AB161 | I／O | アドレスバス | B22 | ＋12V | | |
| A23 | AB171 | 〃 | 〃 | B23 | ＋12V | | |
| A24 | AB181 | 〃 | 〃 | B24 | IR31 | I | INT0 |
| A25 | AB191 | 〃 | 〃 | B25 | IR51 | I | INT1 |
| A26 | AB201 | 〃 | 〃 | B26 | IR61 | I | INT2 |
| A27 | AB211 | 〃 | 〃 | B27 | IR91 | I | INT3 |

| 端子番号 | 信　号　名 | 方　向 | 機　　能 | 端子番号 | 信　号　名 | 方　向 | 機　　能 |
|---|---|---|---|---|---|---|---|
| A28 | AB221 | I／O | アドレスバス | B28 | IR101/IR111 | I | INT41 / INT42 |
| A29 | AB231 | 〃 | 〃 | B29 | IR121 | I | INT5 |
| A30 | INT0 | O | | B30 | IR131 | I | INT6 |
| A31 | GND | | | B31 | GND | | |
| A32 | IOCHK0 | I | 外部NMI | B32 | −12V | | |
| A33 | IOR0 | I／O | コマンド | B33 | −12V | | |
| A34 | IOW0 | I／O | 〃 | B34 | RESET0 | O | RESET |
| A35 | MRC0 | I／O | 〃 | B35 | DACK00/30 | O | HDC |
| A36 | MWC0 | I／O | 〃 | B36 | DACK30/20/10 | O | 640KB/1MBFDC |
| A37 | INTA0 | I／O | 割り込み | B37 | DRQ00/30 | I | HDC |
| A38 | NOWAIT0 | I | | B38 | DRQ30/20/10 | I | 640KB/1MBFDC |
| A39 | SALE1 | I／O | アドレスラッチ | B39 | WORD0 | I | |
| A40 | MACS0 | I | | B40 | EXHRQ10/BANK00 | I | |
| A41 | GND | | | B41 | GND | | |
| A42 | CPUENB0 | O | | B42 | EXHLA10 | O | |
| A43 | RFSH0 | O | リフレッシュ信号 | B43 | DMATC0 | O | END OF PROCESS |
| A44 | BHE0 | I／O | | B44 | NMI0 | O | |
| A45 | IORDY1 | I | レディー信号 | B45 | MWE0 | I／O | |
| A46 | SCLK1 | O | 9.8304/7.9872MHz | B46 | EXHLA20 | O | |
| A47 | S18CLK1 | O | 307.2KHz | B47 | EXHRQ20 | I | |
| A48 | POWER0 | O | 電源確定信号 | B48 | SBUSRQ1 | O | |
| A49 | ＋5V | | | B49 | ＋5V | | |
| A50 | ＋5V | | | B50 | ＋5V | | |

注：次に示す信号はスロットによって異なる．

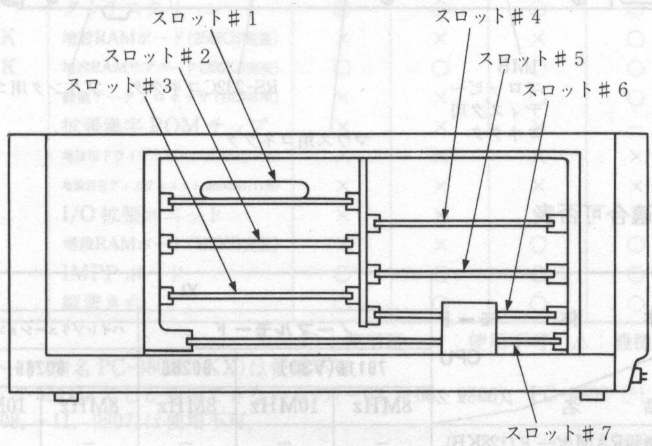
| 端子番号 | 信号名 | PC-98XA | | | | PC-98XL |
|---|---|---|---|---|---|---|
| | | #1 | #2 | #3,5 | #4 | #1～#4 |
| B28 | IR101−INT41 | | ○ | ○ | ○ | ○ |
| | IR111−INT42 (1MBFDC) | ○ | | | | |
| B35 | DACK00 | ○ | ○ | ○ | | ○ |
| | DACK30 | | | | ○ | |
| B36 | DACK10 | ○ | | | | |
| | DACK20 | | ○ | | | |
| | DACK30 | | | ○ | ○ | ○ |
| B37 | DRQ00 | ○ | ○ | ○ | | ○ |
| | DRQ30 | | | | ○ | |
| B38 | DRQ10 | ○ | | | | |
| | DRQ20 | | ○ | | | |
| | DRQ30 | | | ○ | ○ | ○ |
| B40 | EXHRQ10 | | ○ | ○ | ○ | ○ |
| | BANK00 | ○ | | | | |

② ピンコネクション



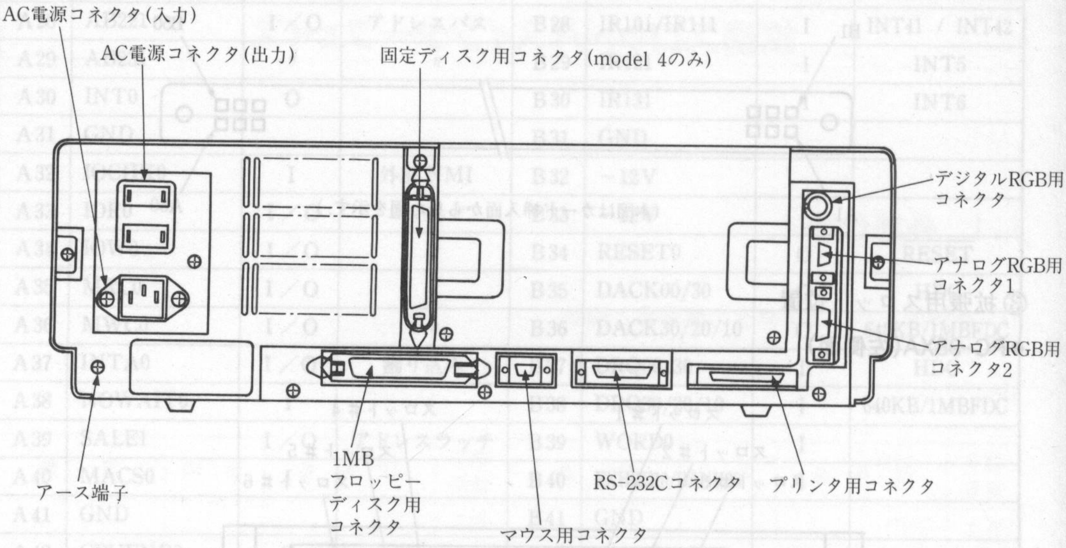

③ 拡張用スロット位置

●PC-98XA(左側面)




・スロット＃1
1MBFDCインターフェイスボード(標準装備)
・スロット＃2
640KBFDCインターフェイスボード(オプション)専用スロット．またはDMAチャネル3．1R10を使わないオプション実装可．
・スロット＃3
オプションスロット
・スロット＃4
固定ディスクインターフェイスボード用スロット(PC-98XAモデル3，31は標準装備)他のモデルの場合，DMAチャネル0未使用オプション実装可(PC-9801-07，PC-9801-27実装可)．
・スロット＃5
オプションスロット
・スロット＃6
CPUスロット(標準装備)
・スロット＃7
テキストVRAMスロット(標準装備)

●PC-98XL(背面)




④ NEC製ボードの適合可否表

| 型番 | 品名 ＼ 本体 / モード / CPU / クロック | XL ノーマルモード 70116(V30) 8MHz | XL ノーマルモード 70116(V30) 10MHz | XL ノーマルモード 80286 8MHz | XL ハイレゾリューションモード 80286 8MHz | XL ハイレゾリューションモード 80286 10MHz | XA 80286 8MHz |
|---|---|---|---|---|---|---|---|
| PC-9801-02L/N | 増設RAMボード(128KB) | — | — | — | — | — | — |
| PC-9801-04 | ユニバーサルボード | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9801-07 | 固定ディスクインターフェイスボード | × | × | × | ○ | ○ | ○ |
| PC-9801-09 | 640KB FDインターフェイスボード | ○ | × | × | ○ | × | ○ |
| PC-9801-13 | CMTインターフェイスボード | × | × | × | × | × | × |
| PC-9801-14 | ミュージックインターフェイスボード | ○ | × | × | × | × | ○ |
| PC-9801-15 | 1MB FDインターフェイスボード | × | × | × | × | × | × |
| PC-9801-16 | 68000ボード | × | × | × | × | × | × |
| PC-9801-17 | 68000用　増設RAMボード | × | × | × | × | × | × |
| PC-9801-19 | GP-IBインターフェイスボード(9801/E/F/M用) | × | × | × | ○ | × | ○ |
| PC-9801-21/K | 増設 RAM サブボード | — | — | — | — | — | — |
| PC-9801-22 | UV/VF/VM用数値データプロセッサ | ○ | × | × | × | × | × |
| PC-9801-23 | 8086ボード | × | × | × | × | × | × |
| PC-9801-24 | 16色グラフィックボード | — | — | — | — | — | — |
| PC-9801-25 | スーパーインポーズボード | ○ | ○ | × | × | × | × |
| PC-9801-26 | サウンドボード | ○ | ○ | × | ○ | ○ | × |
| PC-9801-26K | サウンドボード | ○ | ○ | ○ | ○ | ○ | × |
| PC-9801-27 | 固定ディスクインターフェイスボード | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9801-28 | 拡張漢字 ROM チップ | ○ | ○ | ○ | — | — | — |
| PC-9801-29 | GP-IBインターフェイスボード | ○ | ○ | × | ○ | ○ | ○ |
| PC-9801-29K | GP-IBインターフェイスボード | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9801-31 | 増設RAMボード(10MHz対応) | — | — | — | — | — | — |
| PC-9801-33 | 数値データプロセッサ | ○ | × | × | × | × | × |
| PC-9801-36 | カートリッジ磁気テープユニット用I/Fボード | ○ | ○ | ○ | ○ | ○ | ○ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PC-9801-41 | 増設RAMボード(10MHz,256KB実装) | — | — | — | — | — | — |
| PC-9801-51 | 増設RAMボード(10MHz,1MB実装) | × | × | ○ | ○ | ○ | ○ |
| PC-9805K | 増設 RAM | — | — | — | — | — | — |
| PC-9808 | 数値データプロセッサ(8086CPU 8MHz用) | × | × | × | × | × | × |
| PC-9811K | I/O 拡張ユニット | × | × | × | × | × | × |
| PC-9811N | I/O 拡張ユニット | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9861 | RS-232C拡張インターフェイスボード | ○ | ○ | × | ○ | ○ | ○ |
| PC-9861K | RS-232C拡張インターフェイスボード | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9862 | 通信制御アダプタ | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9863 | モデムボード | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9864 | ネットワークインターフェイスセット | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9864-03 | ネットワーク用 ROM | ○ | ○ | ○ | — | — | — |
| PC-9866 | 通信制御アダプタ | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9872 | マウス | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9872K | マウス | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-9873 | タッチスクリーン | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-98XA-01/K | 増設RAMボード(256KB実装) | × | × | × | ○ | × | ○ |
| PC-98XA-02/K | 増設RAMサブボード(256KB実装) | ○ | ○ | ○ | ○ | × | ○ |
| PC-98XA-03 | 数値データプロセッサ(80286用) | × | × | ○ | ○ | ○ | ○ |
| PC-98XA-04 | 拡張漢字 ROM チップ | × | × | × | — | — | △ |
| PC-98XA-05 | 増設用ドライブユニット(model1/11用) | × | × | × | × | × | △ |
| PC-98XA-06 | 増設固定ディスクユニット(model1/11用) | × | × | × | × | × | △ |
| PC-98XA11 | I/O 拡張ユニット | × | × | × | × | × | ○ |
| PC-98XL-01 | 増設RAMボード(512KB実装) | × | × | ○ | ○ | ○ | × |
| PC-98XA-02 | IMPP ボード | ○ | ○ | ○ | ○ | ○ | ○ |
| PC-98XA-06 | 縦置き台 | ○ | ○ | ○ | ○ | ○ | × |

○：使用可　×：使用不可　△：機種により使用可　－：不要

注：U 2 専用ボード(型名 PC-9801U-XX)は使用不可.
　CPU クロック 5MHz でしか使用できないもの(−03, −06, 9806), PC-9801 でしか使用できないもの(−01, −05, −08, −11, 9807)は使用不可.

### (2) 信号の概要

① AB001～AB191　アドレスバス

アドレスバスは AB001～AB231 の 24 ビットで構成されており，最大 16M バイトのアドレス空間を持つ．ただし，以前からある PC-9800 シリーズ用の ROM 搭載オプションボードでは AB201～AB231 をデコードしていないため，PC-98XL の拡張用スロットの基部にマイクロスイッチ(およびジャンパスイッチ)を設け，旧来のボードが挿入されている場合には，AB201～AB231＝0 の時にのみ AB191 がアクティブになるようにしている．マイクロスイッチ(またはジャンパスイッチ)が ON になっている場合には，AB191 には本来のアドレス信号が出力される．

② BHE0　バスハイ イネーブル

データバスの上位半分にデータを出力することを許可したり，上位半分の側に接続されているデバイスを CPU がアクセスすることを示す信号である．下位半分側に対しては AB001 が同様の働きをする．

③ DB001 ~ DB151　データバス

データバスは DB001 ~ DB151 の 16 ビットで構成されている.

④ IOR0　　IO リード

IO アクセスのリードストローブ信号である. I/O をリードするサイクルに Low になる.

⑤ IOW0　　IO ライト

IO アクセスのライトストローブ信号である. I/O にデータを出力するサイクルに Low になる.

⑥ MRC0　メモリリード

メモリアクセスのリードストローブ信号である. メモリをリードするサイクルに Low になる.

⑦ MWC0　メモリライト

メモリアクセスのライトストローブ信号である. メモリにデータを書き込むサイクルに Low になる.

⑧ MWE0　メモリライトイネーブル

Low の時, 拡張メモリに対して D-RAM の書き込みタイミングを供給する.

⑨ RFSH0　リフレッシュ

Low の時, バスをリフレッシュのために占有することを示す信号である.

⑩ IR31, IR51, IR61(以上, マスタ PIC), IR91, IR101, IR111, IR121, IR131(以上, スレーブ PIC)割り込み要求信号.

要求信号のポジティブエッジによって, CPU に対してマスカブル割り込みをかけることができる. 拡張スロットに接続される入出力装置の割り込みには次表のものがある.

IR91, IR101, IR111, IR131 は各デバイス用にリザーブされている.

IR91 はオープンコレクタ出力としてプルアップ抵抗を使用する. 他はスリーステートバッファを使用する.

| 信号名 | 接続装置 | 備考 |
|---|---|---|
| IR 31 | | INT 0 |
| IR 51 | | INT 1 |
| IR 61 | | INT 2 |
| IR 91 | 固定ディスク | INT 3 |
| IR101 | 640KB フロッピーディスク | INT41 |
| IR111 | 1MB フロッピーディスク | INT42 |
| IR121 | | INT 5 |
| IR131 | マウス | INT 6 |

⑪ IOCHK0　ノンマスカブルインタラプト信号

本信号のネガティブエッジにより，CPU にノンマスカブルの割り込みをかけることができる．

ノンマスカブルインタラプトはソフトウェアからのマスク制御ができないハードウェアからの割り込みである．実際には，拡張メモリパリティエラー(リード時)検出に使用しており，NMI-F/F のセット／リセットによりこの割り込みを許可／禁止できるようになっている．

| コマンド | 機　　　能 |
|---|---|
| OUT 52 | NMI F/F のセット(許可) |
| OUT 50 | 〃　　リセット(禁止) |

なお，システムポート($\mu$PD8255A)の MCKEN が〝1″の時には(NMI F/F の 1，0 にかかわらず)同じシステムポートをリードすることにより EMCK1 または IMCK が 1 であることをチェックすることができる．

⑫ INT0　(マスカブル)インタラプト

インタラプトコントローラ $\mu$PD8259A の INT 出力信号である．

各割り込み要求に $\mu$PD8259A が応答した時にアクティブになる．

⑬ NMI0　(ノンマスカブル)インタラプト

ノンマスカブル割り込みがあった時にアクティブになる信号である．

⑭ SCLK1 System clock

CPU クロック　PC-98XA/XL 8M モード … 7.9872MHz

PC-98XL 10M モード　…… 9.8304MHz

⑮ S18CLK1　(307.2KHz)

クロックは SCLK と S18CLK1(307.2KHz)が供給される．

⑯ POWER0　電源確定信号

電源 ON/OFF 時，DC＋5V 電源出力が＋4.75V 以上になった場合に有効になる．

⑰ RESET0　リセット信号

RESET0 は DC＋5V 電源が 4.75V 以下になるか，本体のリセット SW の押下で有効になる．

⑱ DRQ00，DRQ10，DRQ20，DRQ30(DMA Request)DMA 要求信号

⑲ DACK00，DACK10，DACK20，DACK30(DMA Acknowledege)DMAアクノリッジ信号

⑱，⑲で解説されている信号線とそれを使用するデバイスの対応は，第2部「2章　DMAコントローラ」を参照のこと．

⑳ WORD0　ワード／バイト

内部DMAに接続するI/Oデバイスがワード転送かバイト転送かを示す信号．ワード転送のとき，DACK信号と同期させて，この信号をLowにしなければならない．スリーステート出力とする．

㉑ DMATC0　DMAターミナルカウント

DMA転送の際に，最終ワード／バイトの時，Lowになる．

㉒ CPUENB10　CPUイネーブル信号

CPUがバスを使用しているときにアクティブとなる．

㉓ IORDY1　IOレディ

CPUおよび内部DMAに対するウェイト要求信号．スリーステート出力とする．

CPUがI/Oをアクセスする時，システム内部で自動的にWaitが入る．このバスサイクルで長さが十分であれば，デバイス側で本信号を操作する必要はない．

㉔ GND　：グランド

㉕ +5V　：+5V電源ライン

㉖ +12V：+12V電源ライン

㉗ −12V：−12V電源ライン

㉘ V1，V2：オプション用電源ライン

次にPC-9800シリーズと互換性を持たない信号線(CPUが8086/V30から80286になって追加された)を示す．PC-98XLでは，これらの信号は80286CPU動作時のみ有効となる．

㉙ AB201～AB231 アドレスバス

AB101～AB191は，8086/V30でアクセス可能であるが，それ以上のアドレスバスをアクセスすることが出来ない．80286ではAB101～AB191に加えてAB201～AB231のアドレスバスがアクセス可能である．PC-9800シリーズではAB201～AB231は未接続である．

㉚ EXHRQ10/EXHRQ20　ホールドリクエスト信号

外部CPU/DMAからのバス要求信号．

㉛ EXHLA10/EXHLA20　ホールドアクノリッジ信号

外部 CPU/DMA へのアクノリッジ.

㉜ SBUSREQ1　バス解放要求信号

内部 DMA, リフレッシュ制御回路から外部 CPU/DMA に対するバス解放要求信号.

㉝ NOWAIT0　ノーウェイト信号

メモリを 0Wait で動かすときの要求信号.

㉞ SALE1　上位アドレスバスラッチ信号

アドレスバス AB161 ～ AB231 のラッチ要求信号.

㉟ INTA0　外部 CPU データ要求信号

外部 CPU から 8259PIC に対するデータ要求信号.

㊱ MACS0　メモリボード自己アクセス信号

オプションのメモリボードが, 自身に対してアクセスされていることを示すために出力する信号.

I/O 拡張ユニットで使用する. オープンコレクタ出力とする.

## 1.2 DC特性

### (1) DC特性

| 信号名 | 1スロット当りの最大入力電流 $I_{IL}$ (mA) | 1スロット当りの最大入力電流 $I_{IH}$ ($\mu$A) | 外部ロジックがドライブする最小出力電流(注) $I_{OL}$ (mA) | 外部ロジックがドライブする最小出力電流(注) $I_{OH}$ (mA) |
|---|---|---|---|---|
| AB001～AB231 | | | | |
| BHE0 | | 40 | | |
| DB001～DB151 | | | | |
| IOR0 | | | 1.2 | −1.2 |
| IOW0 | −0.8 | 50 | | |
| MRC0 | | | | |
| NWC0 | | | | |
| MWE0 | | 40 | | |
| RFSH0 | | | 不可 | |
| IR31～IR131 | — | — | 8 | −0.4 |
| IOCHK0 | — | — | | |
| INT0 | | | 不可 | |
| NMI0 | | | — | — |
| SCLK1 | −0.8 | 40 | | |
| S18CLK1 | | | 不可 | |
| POWER0 | | | | |
| RESET0 | | | | |
| DRQ00, DRQ30 | — | — | 8 | −0.4 |
| DACK00, DACK30 | −0.8 | 40 | 不可 | |
| WORD0 | | | 8 | −0.4 |
| DMATC0 | −0.4 | 20 | 不可 | |
| CPUENB10 | −0.8 | 40 | 12 | −1.2 |
| IORDY1 | — | — | 8 | −0.4 |
| EXHRQ10, EXHRQ20 | | — | | |
| EXHLA10, EXHLA20 | −0.8 | 40 | | |
| SBUSRQ1 | | | 不可 | |
| NOWAIT0 | — | — | 8 | −0.4 |
| SALE1 | −0.8 | 50 | 12 | −1.2 |
| INTA0 | | 40 | 8 | −0.4 |
| MACS0 | — | — | | |

注：外部ロジックは，IR91, IOCHK0, DRQ00, DRQ30, EXHRQ10, EXHRQ20, NOWAIT0, MACS0がオープンコレクタで，他はスリーステート出力であること．

(2) 電源容量

| DC | 変動率 | 1スロット当りの容量 | 4(3)スロット合計の最大容量 |
|---|---|---|---|
| +5V | ±5%以内 | 0.8 A | 3.2 A (2.4 A) |
| +12V | ±10%以内 | 0.06A | 0.24A (0.18A) |
| −12V | ±10%以内 | 0.07A | 0.28A (0.21A) |

## 1.3 AC特性

PC-98XL/XAの拡張バスは80286CPUに対して最適化がなされているため，次の点に注意する必要がある．詳細については後述のタイムチャートを参照すること．

(1) I/Oアクセス

PC-9800シリーズの8MHzで動作するオプションは，PC-98XA/XLの8MHzでもそのまま動くよう考慮されている．

ただし，SCLK1のデューティー比が1：1になっている点に注意すること(図6を参照)．

(2) メモリアクセス

PC-98XA/XLハイレゾリューションモード時のバス信号は80286CPUに最適化されているため，PC-9800シリーズのメモリ(RAM/ROM)は動かない．

また，PC-9800シリーズ用オプションをそのまま本システムに実装した場合，メモリパリティエラー，データ化けを起こし，最悪の場合には本体(データアドレスバッファ)が破壊される可能性もあるため注意すること．

NEC純製のオプションボードの場合，オプションボード上にあるROMを動作しないように設定することが可能であるから，PC-98XAの場合，必ずボード上のジャンパスイッチを設定しなおしてROMを動作しないモードにすること．

メモリアクセス時の注意事項は次のとおり．

・アドレスバスの上位7ビットは本体側ではラッチされていないため，メモリ側でアドレスデコードした後，SALE1信号によりラッチすること．

また，自分のアドレスであったときMACS0信号を駆動すること(MACS0は拡張ユニットで使用する)．

・メモリアクセスを0wait(バスサイクル2クロック)で動作させることができる場合，NOWAIT0信号を駆動すること．ただし，メモリとしては80ns以下の高速が要求される．

なお，タイムチャートは図5を参照すること．

メモリボード設計時の参考回路を図1，図2に載せてあるので参考にすること．

### (3) 外部CPUおよび外部DMA

PC-9800シリーズにおいて，外部CPUおよび外部DMAは，SCLKに同期して制御信号を出力する必要がある等の制限があったが，本装置においては，EXHRQ10またはEXHRQ20信号によりバス使用を要求し，EXHLA10またはEXHLA20信号により，許可が与えられた後は，独自のタイミングで動作できるよう考慮されている．

EXHRQ10とEXHRQ20の違いは，EXHRQ20の優先度が高いことを除けば，機能的に同じである．

外部CPUおよび外部DMAを使う場合，次の点に注意すること．

・EXHLA信号が入力された時，本体側の，次の制御信号はスリーステート状態になっているため，任意のタイミングでこれらの信号を駆動すること．ただし，拡張バス上にアドレスを出力した後，SALE，コマンド線(MRC0等)等を出力するタイミングは，本体のタイムチャートに準じて時間をとること．

| | |
|---|---|
| 〈駆動信号〉 | AB001～AB231，BHE0，DB001～DB151，IOR0，IOW0，MRC0，MWC0，MWE0，SALE1， |
| 〈外部CPUのみ〉 | CPUENB0，INTA0 |
| 〈外部DMAのみ〉 | WORD0 |

・SBUSRQ1信号を監視し，1の時は直ちにEXHRQを解除し，バスの使用を中止すること．SBUSRQ1信号は，メモリリフレッシュおよび内部DMAの転送要求がある時1となる．

・外部CPU/DMAはI/O拡張ユニットではサポートされない．

### (4) ROM・RAM回路例

図1，2は0E5000～0E5FFFの4KバイトにROMを載せる場合の回路例である．ここで，ROMを0E5000～0E5FFFまでの間に設定する場合，同じ内容がFE5000～FE5FFFにも見えるように設計しなければならない．また，この範囲内のメモリアクセス時にはMACS0信号は出さないようにする．

図1はROMのアドレス デコード回路である．回路AはAB231～AB171のデコード回路であり，これはアドレスデコード後SALE1信号でラッチする必要がある．回路BはAB161～AB121のデコード回路でありこの信号はラッチしないようにする．

デコード回路の設計で注意しなければならないのは，回路の遅延時間である．CPU80286，8MHz動作時を例にとってみると，AB201～AB231からBNKDEC1までの遅延時間をtab，SALE1からBNKDEC1までの遅延時間をtsb，AB161～AB121からADDEC1までの遅延時間をtaa，MRC0からMRC1までの遅延時間をtmmとすると，

$$tab\,(max) - tmm\,(min) = tabx\,(max),\ tabx\,(max) \leq 105ns \quad \cdots\cdots ①$$

$$tsb\,(max) - tmm\,(min) = tsbx\,(max),\ tsbx\,(max) \leq 35ns \quad \cdots\cdots ②$$

$$taa\,(max) - tmm\,(min) = taax\,(max),\ taax\,(max) \leq 25ns \quad \cdots\cdots ③$$

となるように検討しなければならない．図1ではtab (max)＝83ns, tsb (max)＝30ns, taa (max)＝25ns, tmm (min)＝3nsであるから，

83ns－3ns＝80ns≦105ns
30ns－3ns＝27ns≦ 35ns
25ns－3ns＝22ns≦ 25ns

となり上記条件を満足する．

図2は，ROMデータの出力回路である．ここで注意する点はアドレス確定からデータ出力要求時間までの時間がきびしく，この回路では200nsでアクセス可能なROMを使用する必要があることである．これより遅いROMを使用する場合は一時IORDY1をLOWにしてCPUにウエイトをかける必要がある．IORDY1の使用法は拡張バスの外部仕様(1.1)を参照すること．

注：PC-98XLノーマルモードにおいて0C0000H ~ 0DFFFFHをアクセスする場合には，自動的にWAITが挿入されるため，10MHz動作においても，約300nsのアクセスタイムのROMが使用できる．

RAMを0E5E00 ~ 0E5FFFの範囲に載せる時もROMを載せる時と同様な注意事項が必要となる．ただし，アドレスデコードの遅延時間に関する件はRAMの種類により次のようにする．

① D-RAM

D-RAM回路作成時は，RAS信号，CAS信号の基となる信号を作成する時，ROMの時と同様に遅延時間を考慮して作成しなければならない．

② S-RAM

・図3のように$\overline{CS}$信号の確定が$\overline{WE}$信号の立ち下りに対しセットアップ時間を必要とする場合，$\overline{CS}$信号作成回路の遅延が次の式を満足する必要がある．tab, tsb, taaはROMデコード回路のtab, tsb, taaにそれぞれ対応するものとする．$\overline{WE}$信号に対する$\overline{CS}$のセットアップ時間をtsとすると，

tab (max)－tmm (min)＋ts＝tabx (max), tabx (max)≦105ns ………………………④
tsb (max)－tmm (min)＋ts＝tsbx (max), tsbx (max)≦ 35ns ………………………⑤
taa (max)－tmm (min)＋ts＝taax (max), taax (max)≦ 25ns ………………………⑥

となる．

・図4のように$\overline{CS}$信号の確定が$\overline{WE}$信号の立ち下りに影響されなく，しかもWE信号をアドレスデコード信号でマスクしない場合は，ROMのデコード回路で掲げた①②③の式を満足する必要はない．ただし，$\overline{WE}$信号をアドレス デコード信号でマスクする場合は①②③式を満足しなければならない．

図1 ROMデコード回路例

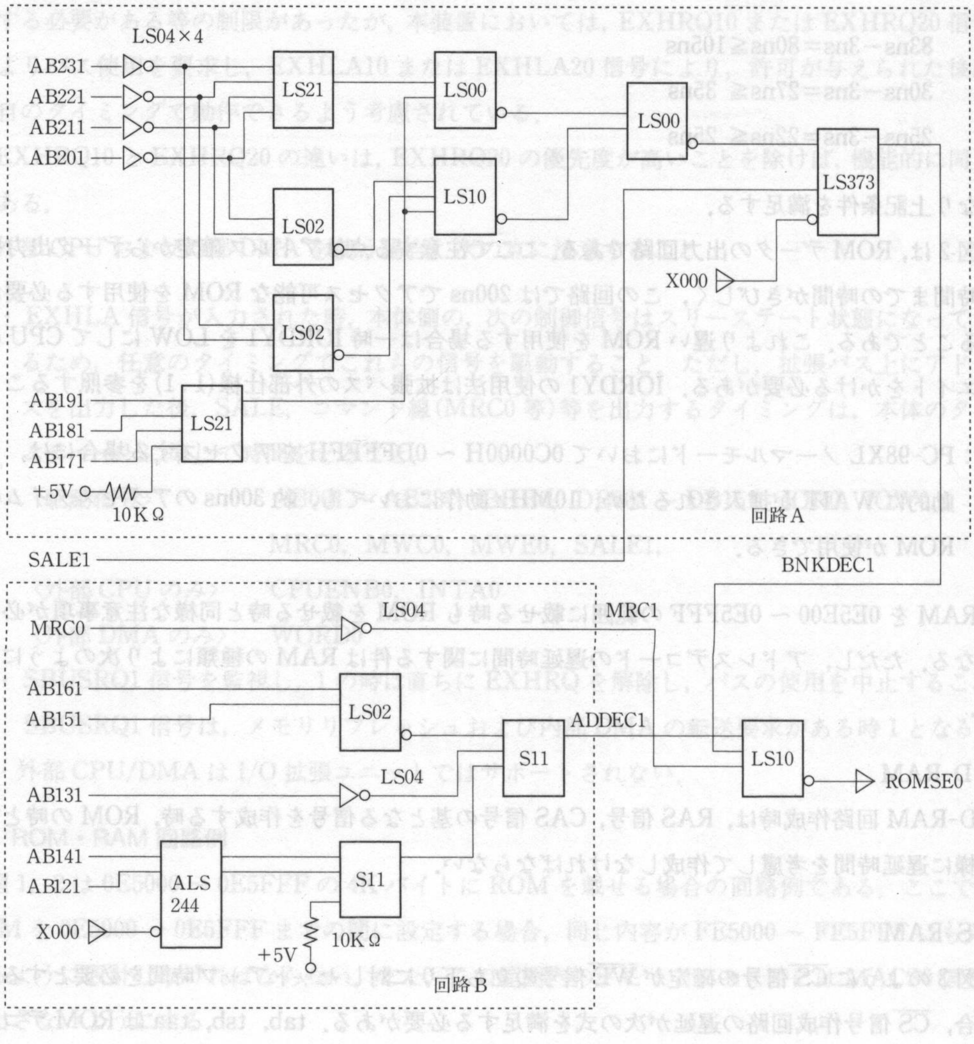
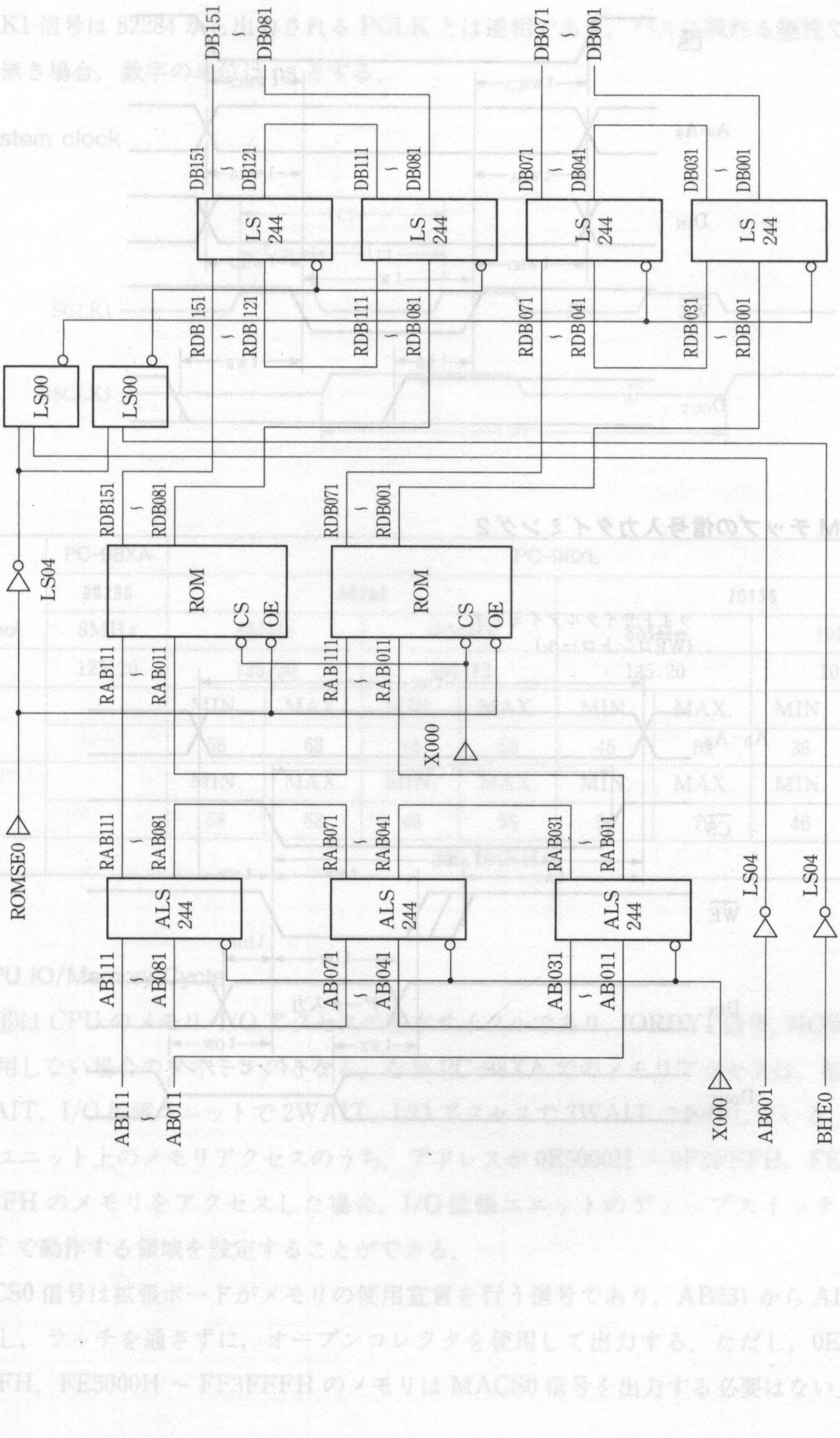
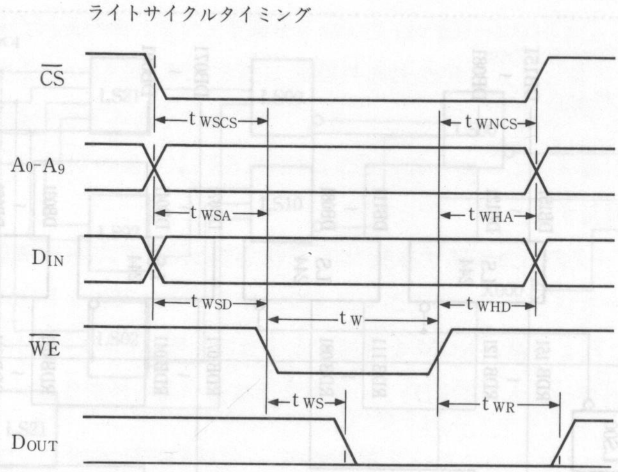


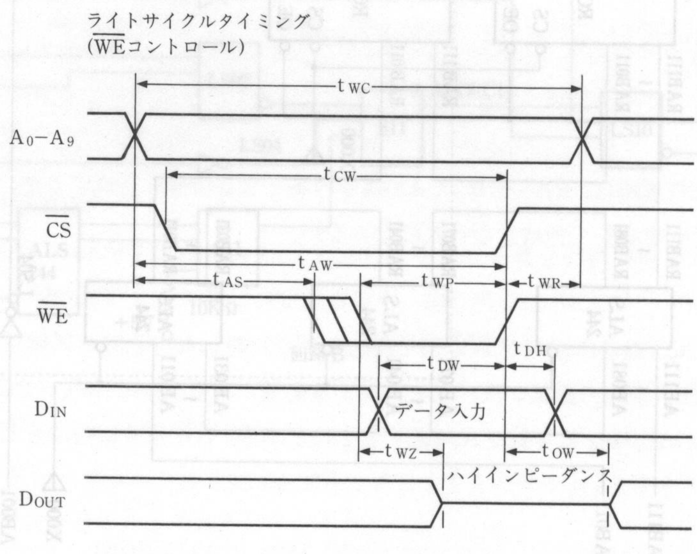
図2　ROM 回路例


DB151 ∫ DB081
DB071 ∫ DB001
DB151 ∫ DB121
DB111 ∫ DB081
DB071 ∫ DB041
DB031 ∫ DB001
LS 244
RDB151 ∫ RDB121
RDB111 ∫ RDB081
RDB071 ∫ RDB041
RDB031 ∫ RDB001
LS00
LS04
RDB151 ∫ RDB081
RDB071 ∫ RDB001
ROM
CS
OE
RAB111 ∫ RAB011
X000
RAB111 ∫ RAB081
RAB071 ∫ RAB041
RAB031 ∫ RAB011
ROMSE0
ALS 244
AB111 ∫ AB081
AB071 ∫ AB041
AB031 ∫ AB011
AB111 ∫ AB011
AB001
BHE0

## 図3 RAM チップの信号入力タイミング1




## 図4 RAM チップの信号入力タイミング2



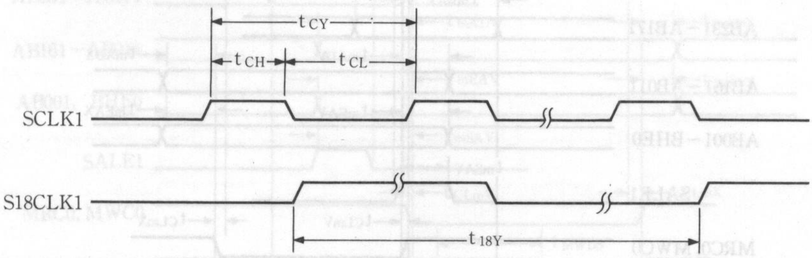
## 1.4 タイミングチャート

＊印の信号は入力タイミングを示しており，値は入力規格値である．

SCLK1 信号は 82284 から出力される PCLK とは逆相であり，バスに現れる極性である．

別記無き場合，数字の単位は ns とする．

### (1) System clock




| Symbol | PC-98XA | PC-98XL | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 80286 | 80286 | | | | 70116 | | | |
| | 8MHz | 8MHz | | 10MHz | | 8MHz | | 10MHz | |
| $t_{CY}$ | 125.20 | 125.20 | | 101.73 | | 125.20 | | 101.73 | |
| $t_{CH}$ | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. |
| | | 58 | 68 | 46 | 55 | 46 | 69 | 38 | 56 |
| $t_{CL}$ | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. |
| | | 58 | 68 | 46 | 55 | 56 | 79 | 46 | 64 |
| $t_{18Y}$ | 307.20(KHz) | | | | | | | | |

### (2) CPU IO/Memory Cycle

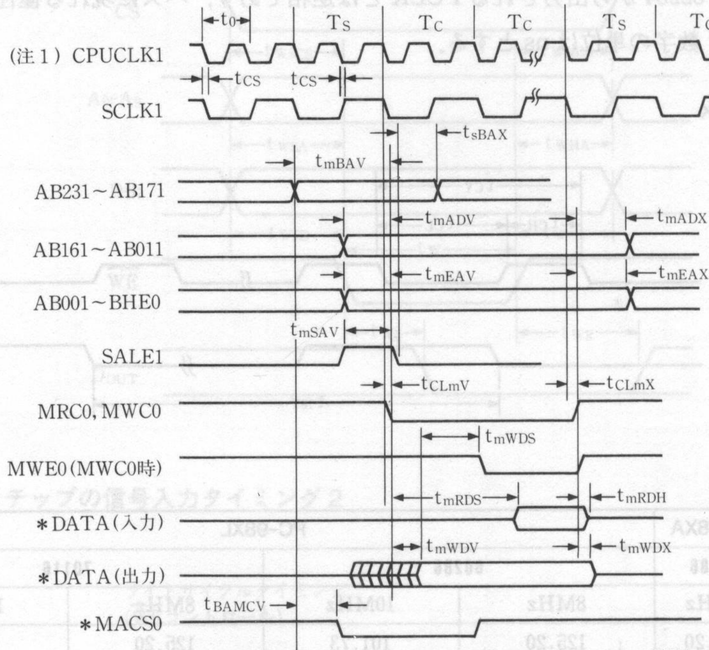
①～⑥は CPU のメモリ·I/O アクセスの基本サイクルであり，IORDY1 信号，NOWAIT0 信号を使用しない場合のタイミングとなる．なお PC-98XA でのメモリアクセスは，拡張バス上で 1WAIT，I/O 拡張ユニットで 2WAIT，I/O アクセスで 3WAIT で動作している．また，I/O 拡張ユニット上のメモリアクセスのうち，アドレスが 0E5000H ～ 0F3FFFH，FE5000H ～ FF3FFFH のメモリをアクセスした場合，I/O 拡張ユニットのディップスイッチによって 3WAIT で動作する領域を設定することができる．

MACS0 信号は拡張ボードがメモリの使用宣言を行う信号であり，AB231 から AB171 をデコードし，ラッチを通さずに，オープンコレクタを使用して出力する．ただし，0E5000H ～ 0F3FFFH，FE5000H ～ FF3FFFH のメモリは MACS0 信号を出力する必要はない．

① 80286 Memory Cycle

PC-98XA

PC-98XL：100000H ～ DFFFFFH

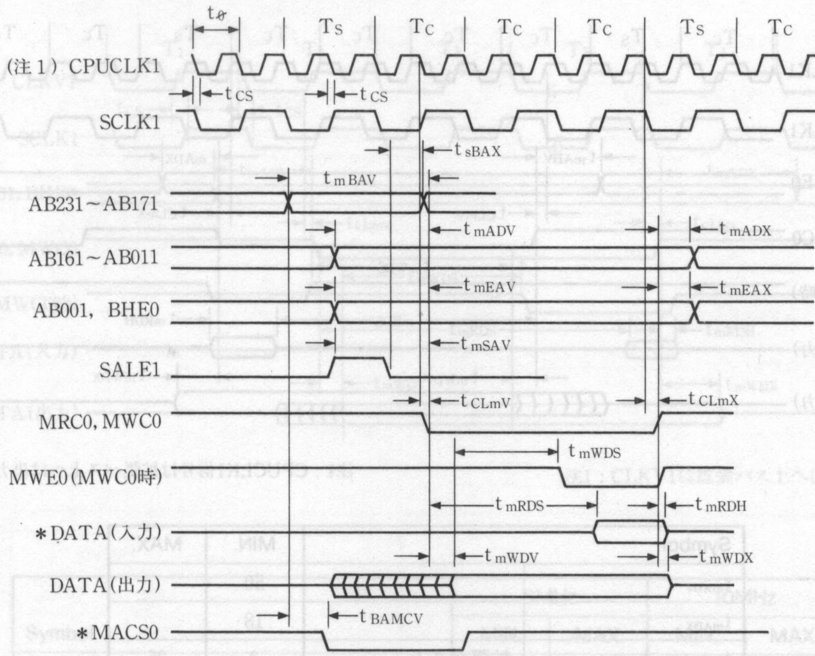



注1：CPUCLK1信号は拡張バス上へは出力されない。

| Symbol | | PC-98XA 8MHz MIN. | PC-98XA 8MHz MAX. | PC-98XL 8MHz MIN. | PC-98XL 8MHz MAX. | PC-98XL 10MHz MIN. | PC-98XL 10MHz MAX. |
|---|---|---|---|---|---|---|---|
| $t_{CS}$ | 拡張バス上 | 4 | 14 | 4 | 14 | 4 | 14 |
| | I/O拡張ユニット上 | | | | | | |
| $t_{mBAV}$ | | 108 | | 101 | | 83 | |
| $t_{sBAX}$ | | 18 | | 18 | | 30 | |
| $t_{mADV}$ | | 27 | | 27 | | 23 | |
| $t_{mEAV}$ | | 23 | | 23 | | 23 | |
| $t_{mSAV}$ | | 37 | | 37 | | 30 | |
| $t_{CLmV}$ | 拡張バス上 | 3 | 25 | 3 | 25 | 3 | 21 |
| | I/O拡張ユニット上 | | | | | | |
| $t_{CLmX}$ | 拡張バス上 | | | 3 | 25 | 3 | 20 |
| | I/O拡張ユニット上 | | | | | | |
| $t_{mADX}$, $t_{mEAX}$ | | 18 | | 18 | | 3 | |
| $t_{mRDS}$ | | | 203 | | 203 | | 160 |
| $t_{mRDH}$ | | 5 | | 5 | | 5 | |
| $t_{mWDS}$ | | 136 | | 120 | | 110 | |
| $t_{mWDV}$ | | | 45 | | 45 | | －4 |
| $t_{mWDX}$ | | 15 | | 15 | | 32 | |
| $t_{BAMCV}$ | | | 72 | | 65 | | |

② 80286 Memory Cycle

PC-98XL : 100000H ~ DFFFFFH 2wait

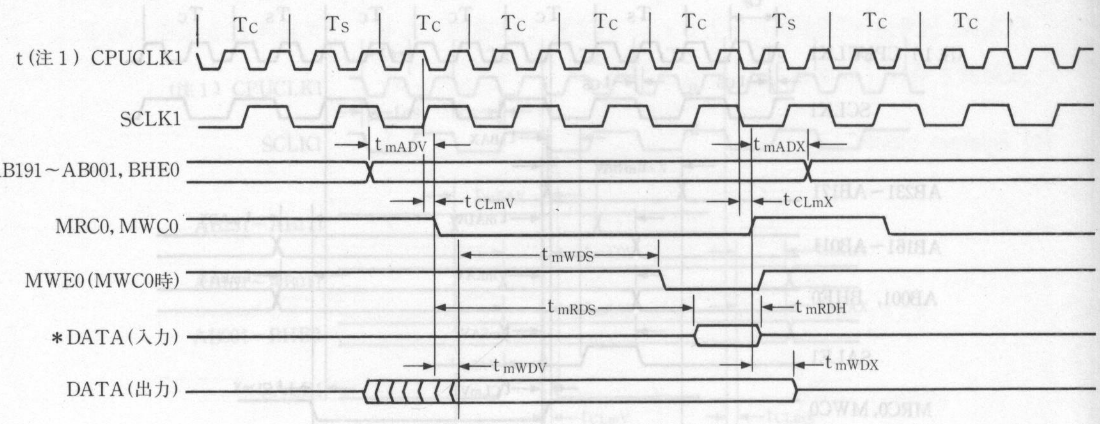



注1 : CPUCLK1信号は拡張バス上へは出力されない.

| Symbol | | 10MHz MIN. | 10MHz MAX. |
|---|---|---|---|
| $t_{CS}$ | 拡張バス上 | 4 | 14 |
| | I/O 拡張ユニット上 | | |
| $t_{mBAV}$ | | 135 | |
| $t_{sBAX}$ | | 30 | |
| $t_{mADV}$ | | 74 | |
| $t_{mEAV}$ | | 74 | |
| $t_{mSAV}$ | | 81 | |
| $t_{CLmV}$ | 拡張バス上 | 3 | 21 |
| | I/O 拡張ユニット上 | | |
| $t_{CLmX}$ | 拡張バス上 | 3 | 20 |
| | I/O 拡張ユニット上 | | |
| $t_{mADX}$, $t_{mEAX}$ | | 3 | |
| $t_{mRDS}$ | | | 211 |
| $t_{mRDH}$ | | 5 | |
| $t_{mWDS}$ | | 161 | |
| $t_{mWDV}$ | | | −55 |
| $t_{mWDX}$ | | 32 | |
| $t_{BAMCV}$ | | | |

### ③ 80286 Memory Cycle

PC-98XL : 0C0000H ~ 0DFFFFH,  FC0000H ~ FDFFFFH

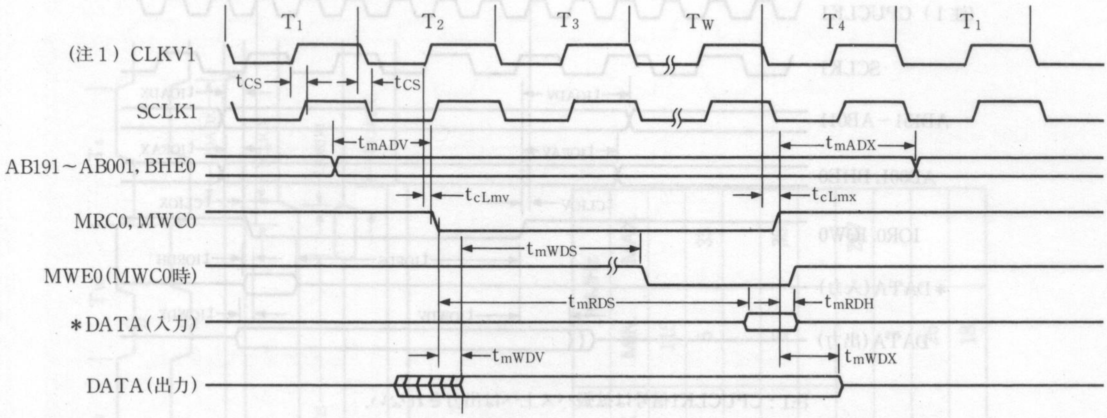



注1 : CPUCLK1信号は拡張バス上へは出力されない.

| Symbol | | MIN. | MAX. |
|---|---|---|---|
| $t_{mADV}$ | | 50 | |
| $t_{mADX}$ | | 18 | |
| $t_{CLmV}$ | 拡張バス上 | 3 | 25 |
| | I/O 拡張ユニット上 | | |
| $t_{CLmX}$ | 拡張バス上 | 3 | 25 |
| | I/O 拡張ユニット上 | | |
| $t_{mWDS}$, $t_{mEAX}$ | | 302 | |
| $t_{mRDS}$ | | | 308 |
| $t_{mRDH}$ | | 5 | |
| $t_{mWDV}$ | | | −17 |
| $t_{mWDX}$ | | 15 | |

④ 70116 Memory Cycle

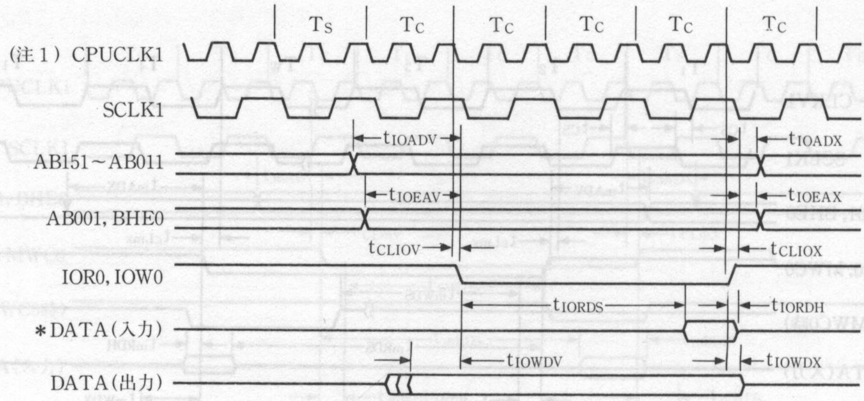
PC-98XL : 0C0000H ~ 0DFFFFH




注1 : CLKV1は拡張バス上へは出力されない.

| Symbol | | 8MHz MIN. | 8MHz MAX. | 10MHz MIN. | 10MHz MAX. |
|---|---|---|---|---|---|
| $t_{CS}$ | 拡張バス上 | 4 | 13 | 4 | 13 |
| | I/O 拡張ユニット上 | | | | |
| $t_{mADV}$ | AB191 | 45 | | 30 | |
| | 上記以外 | 62 | | 50 | |
| $t_{mADX}$ | I/O 拡張ユニット上 | | | | |
| $t_{CLmV}$ | 拡張バス上 | 5 | 33 | 5 | 33 |
| | I/O 拡張ユニット上 | | | | |
| $t_{CLmX}$ | 拡張バス上 | 5 | 33 | 5 | 33 |
| | I/O 拡張ユニット上 | | | | |
| $t_{mWDS}$ | | 310 | | 330 | |
| $t_{mRDS}$ | | | 300 | | 310 |
| $t_{mRDH}$ | | 3 | | 3 | |
| $t_{mWDV}$ | | −1 | | 10 | |
| $t_{mWDX}$ | | 23 | | 15 | |

## ⑤ 80286 IO Cycle

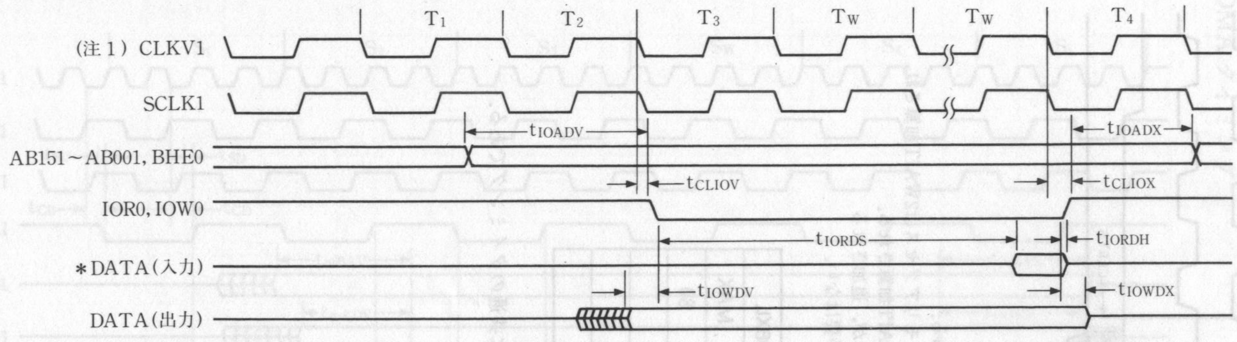



注1：CPUCLK1信号は拡張バス上へは出力されない.

| Symbol | | PC-98XL/XA 8MHz MIN. | PC-98XL/XA 8MHz MAX. | PC-98XL 10MHz MIN. | PC-98XL 10MHz MAX. |
|---|---|---|---|---|---|
| $t_{IOADV}$ | | 115 | | 150 | |
| $t_{IOEAV}$ | | 111 | | 150 | |
| $t_{IOADX}$ | | 25 | | 32 | |
| $t_{IOEAX}$ | | 25 | | 32 | |
| $t_{CLIOV}$ | 拡張バス上 | 3 | 25 | 3 | 21 |
| | I/O 拡張ユニット上 | | | | |
| $t_{CLIOX}$ | 拡張バス上 | 3 | 25 | 3 | 20 |
| | I/O 拡張ユニット上 | | | | |
| $t_{IORDS}$ | | | 270 | | 312 |
| $t_{IORDH}$ | | 5 | | 5 | |
| $t_{IOWDV}$ | | 67 | | 151 | |
| $t_{IOWDX}$ | | 23 | | 32 | |

⑥ 70116 IO Cycle

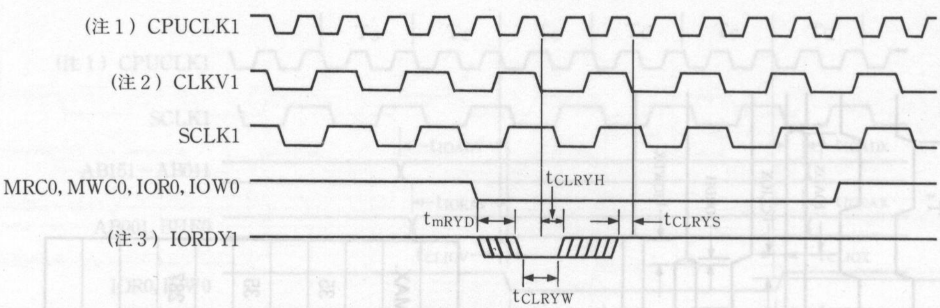
PC-98XL




注1：CLKV1は拡張バス上へは出力されない．

| Symbol | | 8MHz MIN. | 8MHz MAX. | 10MHz MIN. | 10MHz MAX. |
|---|---|---|---|---|---|
| $t_{IOADV}$ | | 150 | | 115 | |
| $t_{CLIOV}$ | 拡張バス上 | 5 | 35 | 5 | 35 |
| | I/O 拡張ユニット上 | | | | |
| $t_{CLIOX}$ | 拡張バス上 | 5 | 35 | 5 | 35 |
| | I/O 拡張ユニット上 | | | | |
| $t_{IORDS}$ | | | 260 | | 287 |
| $t_{IORDH}$ | | 3 | | 3 | |
| $t_{IOWDV}$ | | 48 | | 26 | |
| $t_{IOWDX}$ | | 26 | | 18 | |

## ⑦ CPU IORDY　入力タイミング




注1：80286動作時．この信号は拡張バスへ出力されない．
注2：70116動作時．この信号は拡張バスへ出力されない．
注3：この信号は入力タイミングを示しており，値は入力規格値である．
図のタイミングでIORDY1に信号を入力すると，拡張バスのメモリアクセスは2WAIT加算され，I／Oアクセス，I／O拡張ユニット上のメモリアクセスには1WAIT加算される．
IORDY1の立ち下がりは必ず図のタイミングを守る必要があるが，追加されるWAIT数が不足でよければ，立ち上がりのタイミングは守る必要はない．

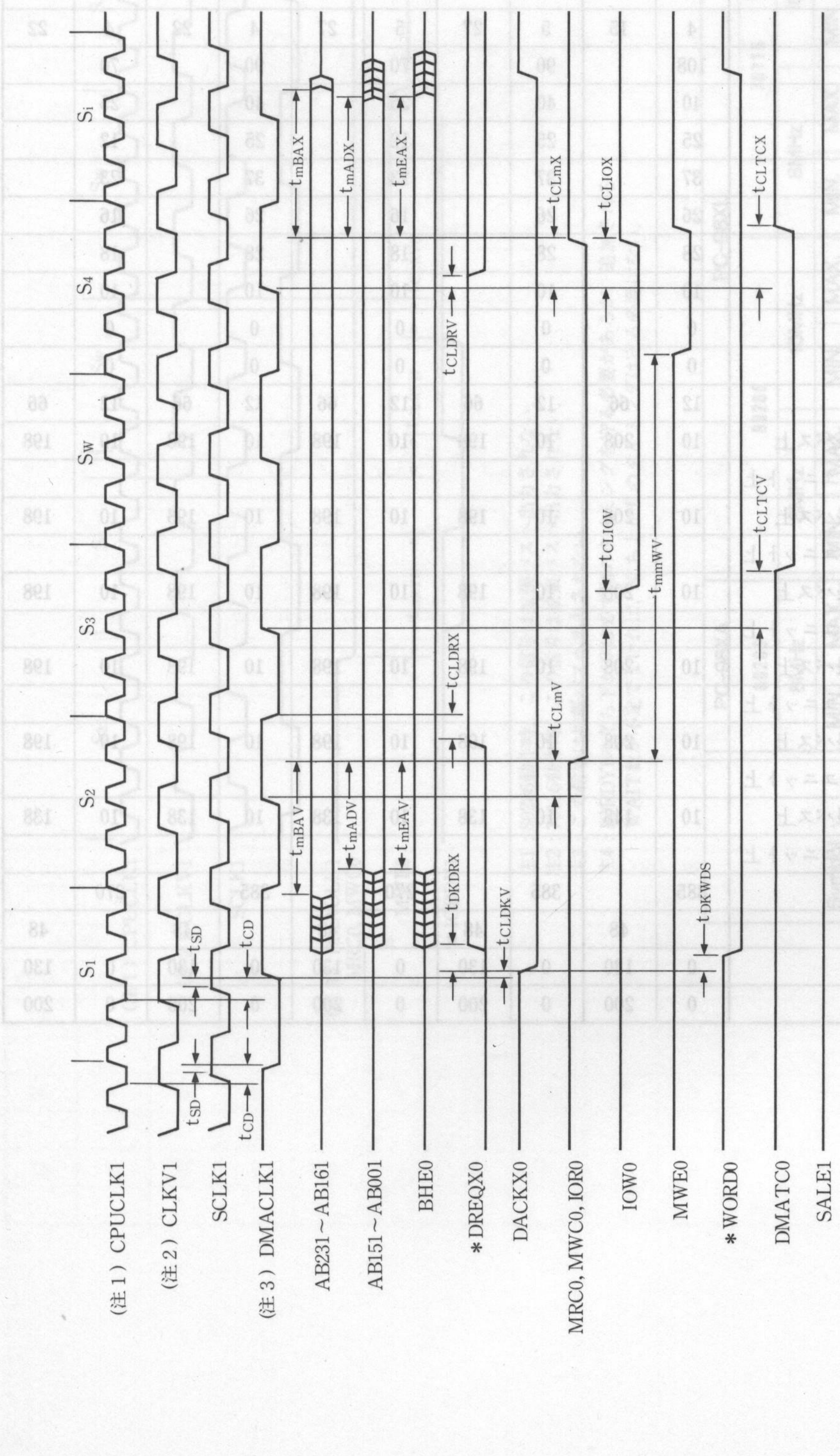
| Symbol | PC-98XA | | PC-98XL | |
|---|---|---|---|---|
| | MIN. | MAX. | MIN. | MAX. |
| $t_{mRYD}$ | | 80 | | 80 |
| $t_{CLRYH}$ | 11 | | 8 | |
| $t_{CLRYS}$ | 37 | | 30 | |
| $t_{CLRYW}$ | 40 | | 40 | |

注4：PC-98XLは，70116／80286，10MHz/8MHzすべてに共通のタイミングである．

### (3) 内部DMAタイミング

DREQX0信号はオープンコレクタで出力し，1KΩ～4KΩ程度のプルアップ抵抗を用いる．WORD0信号はワード転送を行う時のみ使用し，スリーステートで出力する．

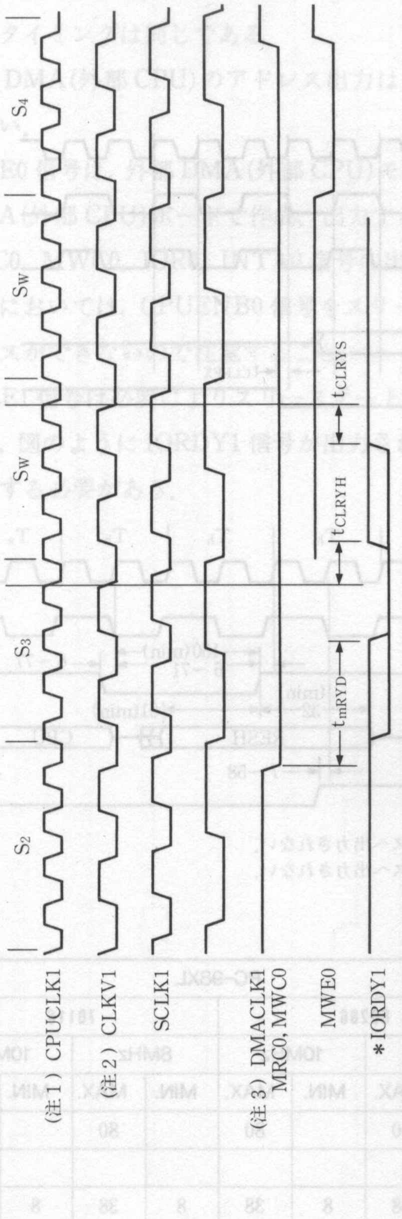
①DMAタイミング




注1：80286動作時，この信号は拡張バスへ出力されない．
注2：70116動作時，この信号は拡張バスへ出力されない．
注3：この信号は拡張バスへ出力されない．

| Symbol | | PC-98XA | | PC-98XL | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 80286 | | 80286 | | | | 70116 | | | |
| | | 8MHz | | 8MHz | | 10MHz | | 8MHz | | 10MHz | |
| | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. |
| $t_{SD}$ | 拡張バス上 | －5 | 6 | －9 | 23 | －9 | 23 | －9 | 18 | －9 | 18 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CD}$ | | 4 | 15 | 5 | 27 | 5 | 27 | 4 | 22 | 4 | 22 |
| $t_{mBAV}$ | | 108 | | 90 | | 70 | | 90 | | 70 | |
| $t_{mADV}$ | | 40 | | 40 | | 25 | | 40 | | 25 | |
| $t_{mEAV}$ | | 25 | | 25 | | 12 | | 25 | | 12 | |
| $t_{mBAX}$ | | 37 | | 37 | | 23 | | 37 | | 23 | |
| $t_{mADX}$ | | 26 | | 26 | | 16 | | 26 | | 16 | |
| $t_{mEAX}$ | | 28 | | 28 | | 18 | | 28 | | 18 | |
| $t_{DKDRX}$ | | 10 | | 10 | | 10 | | 10 | | 10 | |
| $t_{CLDRX}$ | | 0 | | 0 | | 0 | | 0 | | 0 | |
| $t_{CLDRV}$ | | 0 | | 0 | | 0 | | 0 | | 0 | |
| $t_{CLDKV}$ | | 12 | 66 | 12 | 66 | 12 | 66 | 12 | 66 | 12 | 66 |
| $t_{CLmV}$ MRC0 MWC0 | 拡張バス上 | 10 | 208 | 10 | 198 | 10 | 198 | 10 | 198 | 10 | 198 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLmV}$ IOR0 | 拡張バス上 | 10 | 208 | 10 | 198 | 10 | 198 | 10 | 198 | 10 | 198 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLmX}$ MRC0 MWC0 | 拡張バス上 | 10 | 208 | 10 | 198 | 10 | 198 | 10 | 198 | 10 | 198 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLmX}$ IOR0 | 拡張バス上 | 10 | 208 | 10 | 198 | 10 | 198 | 10 | 198 | 10 | 198 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLIOV}$ | 拡張バス上 | 10 | 208 | 10 | 198 | 10 | 198 | 10 | 198 | 10 | 198 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLIOX}$ | 拡張バス上 | 10 | 148 | 10 | 138 | 10 | 138 | 10 | 138 | 10 | 138 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{mmWV}$ | | 385 | | 385 | | 270 | | 385 | | 270 | |
| $t_{DKWDS}$ | | | 48 | | 48 | | 48 | | 48 | | 48 |
| $t_{CLTCV}$ | | 0 | 130 | 0 | 130 | 0 | 130 | 0 | 130 | 0 | 130 |
| $t_{CLTCX}$ | | 0 | 200 | 0 | 200 | 0 | 200 | 0 | 200 | 0 | 200 |

②DMA IORDY1 入力タイミング



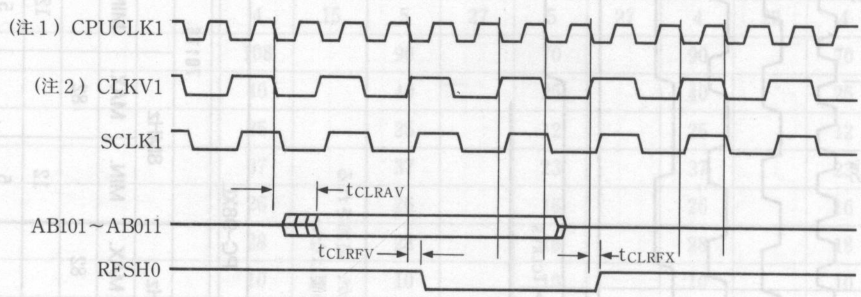

注1：80286動作時，この信号は拡張バスへ出力されない．
注2：70116動作時，この信号は拡張バスへ出力されない．
注3：この信号は拡張バスへ出力されない．
注4：IORDY1の立ち下がりは必ず図のタイミングを守る必要があるが，追加されるWAIT数が不定でよければ，立ち上がりのタイミングは守る必要はない．

| Symbol | PC-98XA 80286 8MHz MIN. | MAX. | PC-98XL 80286 8MHz MIN. | MAX. | 80286 10MHz MIN. | MAX. | 70116 8MHz MIN. | MAX. | 70116 10MHz MIN. | MAX. |
|---|---|---|---|---|---|---|---|---|---|---|
| $t_{mRYD}$ | | 87 | | 82 | | 82 | | 82 | | 82 |
| $t_{CLRYH}$ | 8 | | 16 | | 16 | | 12 | | 12 | |
| $t_{CLRYS}$ | 39 | | 4 | | 4 | | 5 | | 5 | |

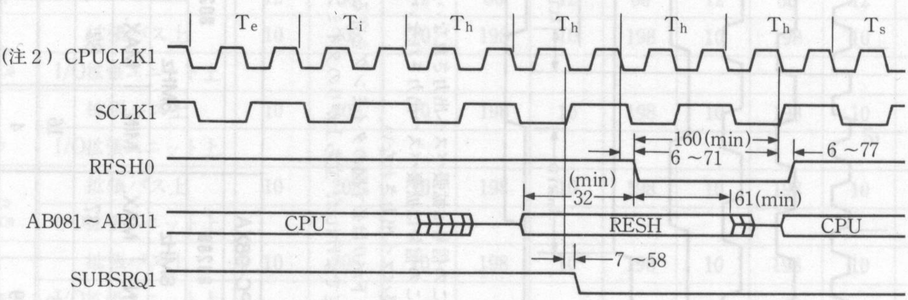
## (4) リフレッシュタイミング

PC-98XL/XA の RFSH0 記号は，PC-9800 シリーズのようなアドレスイネーブル信号ではなく，これ自体がストローブ信号となっている．

● PC-98XL




● PC-98XA




注1：80286動作時．この信号は拡張バスへ出力されない．
注2：70116動作時．この信号は拡張バスへ出力されない．

| Symbol | | PC-98XA 80286 8MHz MIN. | MAX. | PC-98XL 80286 8MHz MIN. | MAX. | 10MHz MIN. | MAX. | 70116 8MHz MIN. | MAX. | 10MHz MIN. | MAX. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_{CLRAV}$ | 拡張バス上 | | | | 80 | | 80 | | 80 | | 80 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLRFV}$ | 拡張バス上 | 6 | 71 | 8 | 38 | 8 | 38 | 8 | 38 | 8 | 38 |
| | I/O拡張ユニット上 | | | | | | | | | | |
| $t_{CLRFX}$ | 拡張バス上 | 6 | 77 | 8 | 37 | 8 | 37 | 8 | 37 | 8 | 37 |
| | I/O拡張ユニット上 | | | | | | | | | | |

### (5) 外部DMAタイミング

外部DMA(外部CPU)はI/O拡張ユニット上では動作不可能である.

ここではEXHRQ20, EXHLA20で示しているがEXHRQ10, EXHLA10も優先順位が違うだけでタイミングは同じである.

外部DMA(外部CPU)のアドレス出力はEXHLA20(10)がLOWの間に行なわれなければならない.

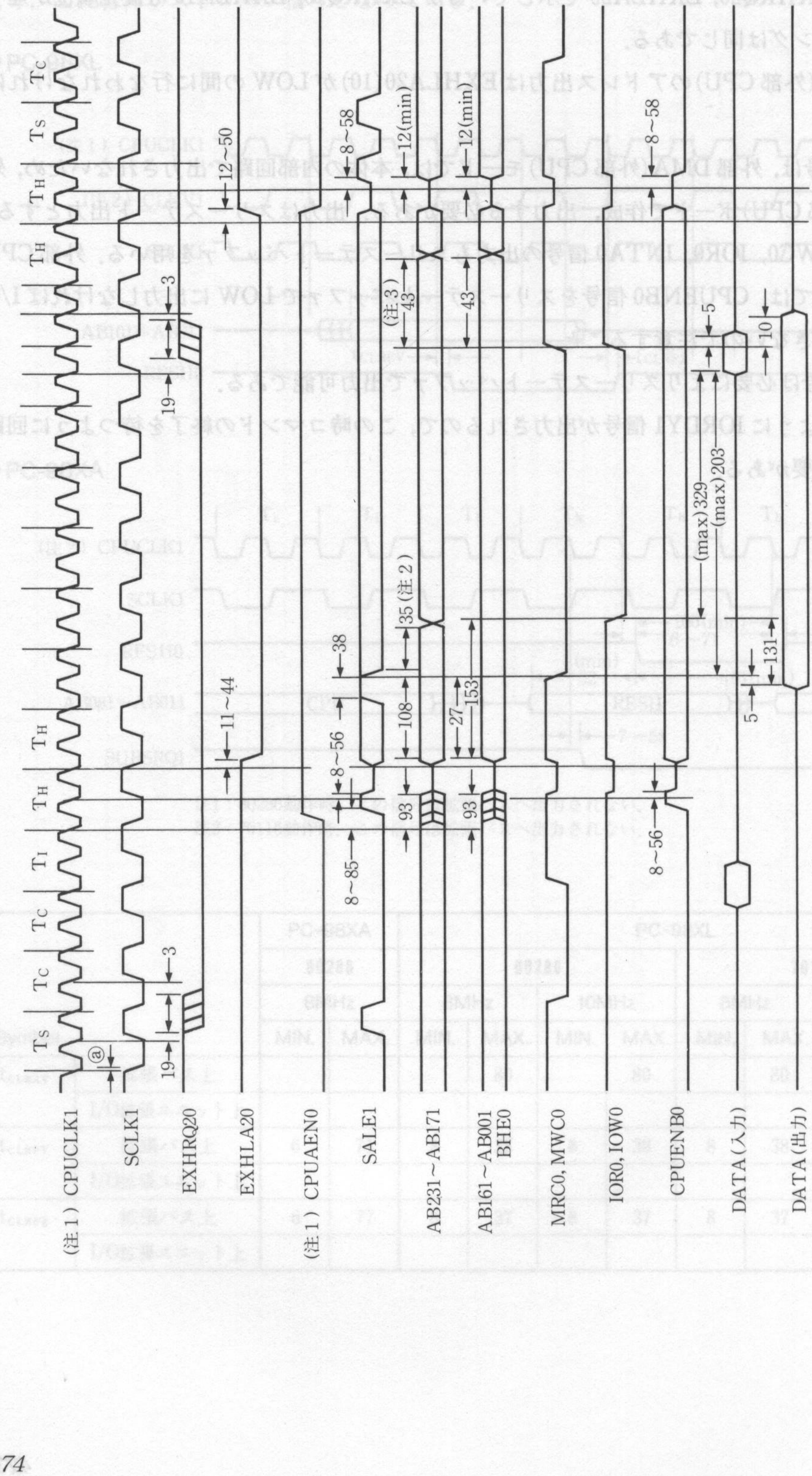
MWE0信号は, 外部DMA(外部CPU)モードでは, 本体の内部回路で出力されないため, 外部DMA(外部CPU)ボードで作成, 出力する必要がある. 出力はスリーステート出力とする.

MRC0, MWC0, IOR0, INTA0信号の出力もスリーステートバッファを用いる. 外部CPUモードにおいては, CPUENB0信号をスリーステートバッファでLOWに出力しなければI/Oアクセスができないので注意すること.

SALE1信号は必要によりスリーステートバッファで出力可能である.

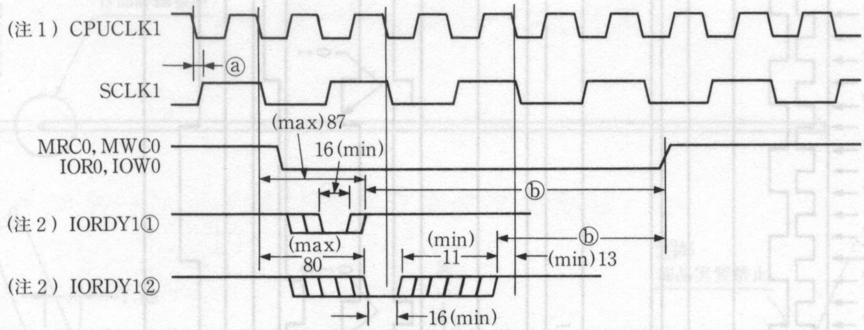
また, 図のようにIORDY1信号が出力されるので, この時コマンドの終了を待つように回路を作成する必要がある.

① 拡張バス上における外部 DMA の動作タイミング




注1：CPUCLK1,CPUAEN0信号は拡張バス上へ出力されない。
注2：SALE1を途中で"0"にする時。
注3：SALE1を"1"のままにする時。

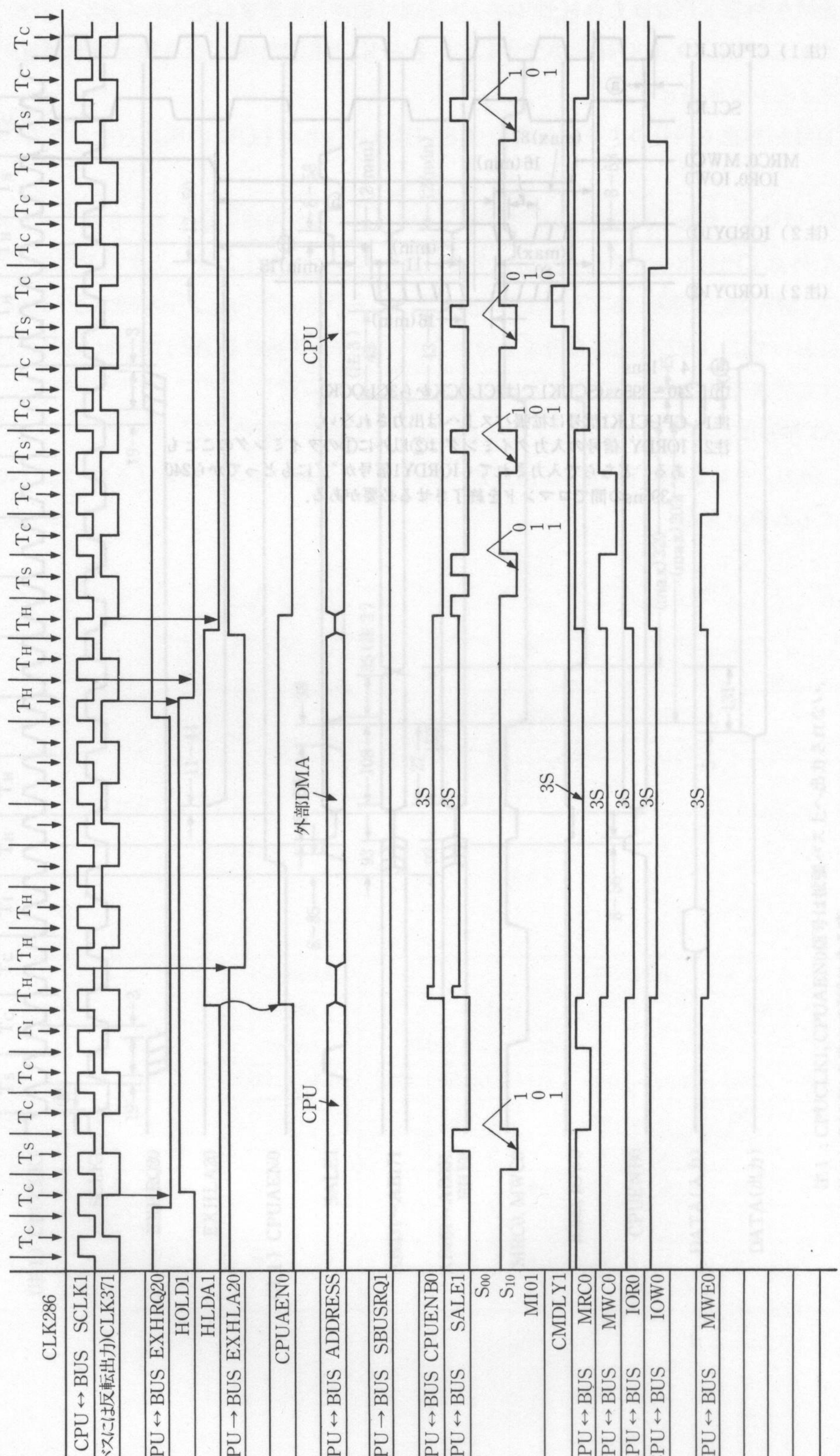
## ② 外部DMA時のIORDY1信号のタイミング



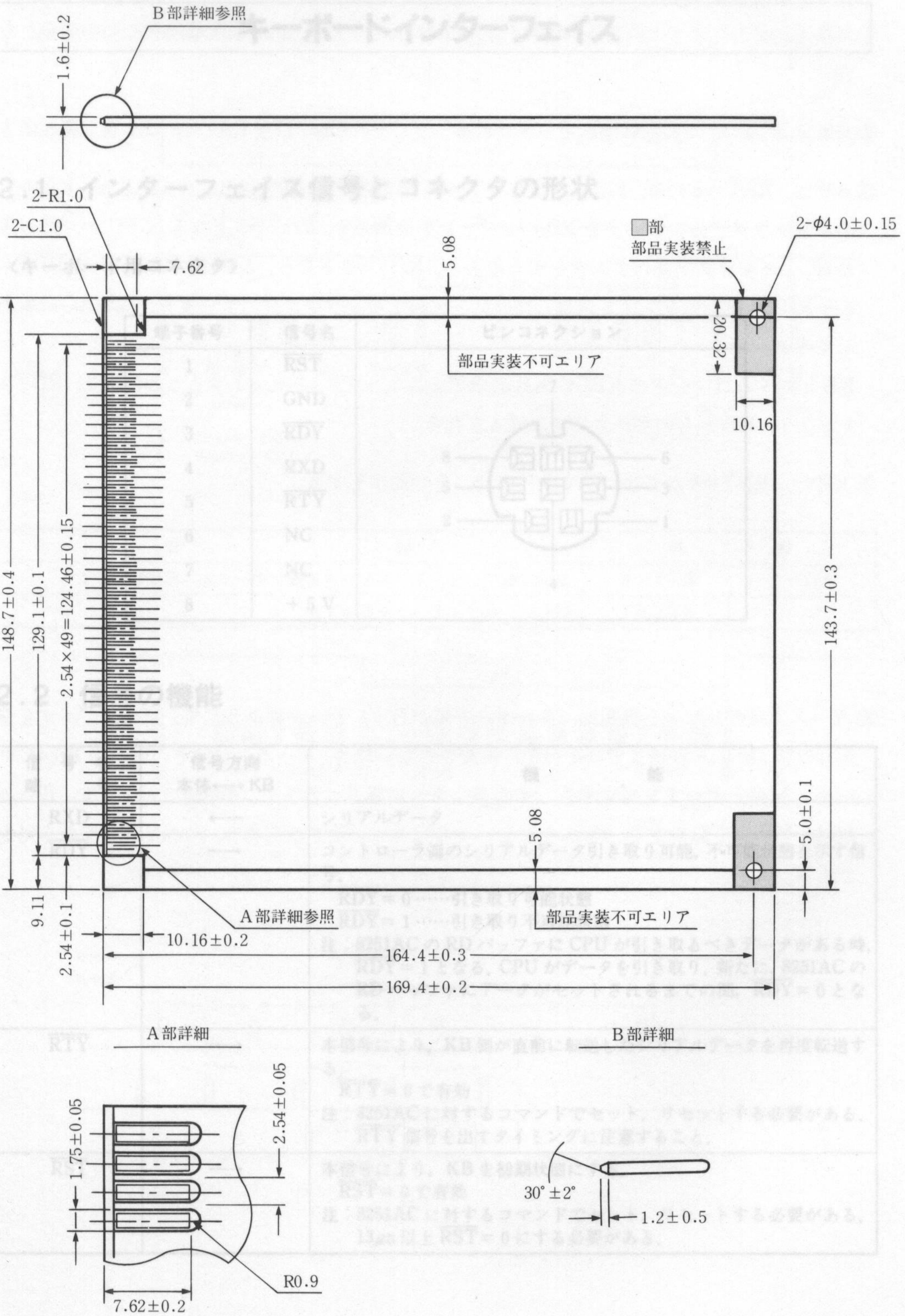

ⓐ　4～14ns
ⓑ　240～395ns(SCLK1では2CLOCKから3CLOCK)

注1：CPUCLK1信号は拡張バス上へは出力されない．
注2：IORDY1信号の入力タイミングは②以外に①のタイミングのこともある．どちらで入力されてもIORDY1信号が"1"にもどってから240～395nsの間でコマンドを終了させる必要がある．

### ③ 外部 DMA の動作シーケンス



## 1.5　ユニバーサルボード外形寸法

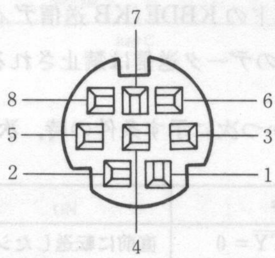


# 第2章

# キーボードインターフェイス

## 2.1 インターフェイス信号とコネクタの形状

〈キーボード用コネクタ〉

| 端子番号 | 信号名 | ピンコネクション |
|---|---|---|
| 1 | $\overline{\text{RST}}$ |  |
| 2 | GND | |
| 3 | $\overline{\text{RDY}}$ | |
| 4 | RXD | |
| 5 | $\overline{\text{RTY}}$ | |
| 6 | NC | |
| 7 | NC | |
| 8 | ＋5 V | |

## 2.2 信号の機能

| 信号名略号 | 信号方向 本体←→KB | 機能 |
|---|---|---|
| RXD | ← | シリアルデータ |
| $\overline{\text{RDY}}$ | → | コントローラ側のシリアルデータ引き取り可能，不可能状態を示す信号．<br>$\overline{\text{RDY}}$＝0……引き取り可能状態<br>$\overline{\text{RDY}}$＝1……引き取り不可能状態<br>注：8251ACのRDバッファにCPUが引き取るべきデータがある時，$\overline{\text{RDY}}$＝1となる．CPUがデータを引き取り，新たに，8251ACのRDバッファにデータがセットされるまでの間，$\overline{\text{RDY}}$＝0となる． |
| $\overline{\text{RTY}}$ | → | 本信号により，KB側が直前に転送したシリアルデータを再度転送する．<br>$\overline{\text{RTY}}$＝0で有効<br>注：8251ACに対するコマンドでセット，リセットする必要がある．$\overline{\text{RTY}}$信号を出すタイミングに注意すること． |
| $\overline{\text{RST}}$ | → | 本信号により，KBを初期状態にする．<br>$\overline{\text{RST}}$＝0で有効<br>注：8251ACに対するコマンドでセット，リセットする必要がある．13μs以上$\overline{\text{RST}}$＝0にする必要がある． |

## 2.3　キーボードの動作

① KBインターフェイスのDATAラインは，KBリセット後，18$\mu$s以上経過しないと確定しない．

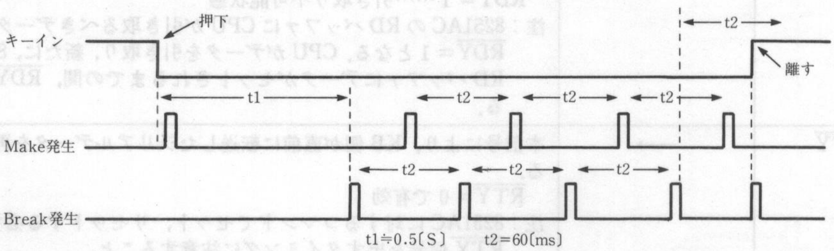
② KBは$\overline{\mathrm{RST}}$=0で初期状態にセットされる．ただし，13$\mu$s以上$\overline{\mathrm{RST}}$=0にする必要がある．

③ KBは，$\overline{\mathrm{RDY}}$=0の時，8251ACにシリアルデータを送信可能である．8251ACが，KBからの1キャラクタ分のデータをRDバッファに引き取ると，$\overline{\mathrm{RDY}}$=1となる．CPUが，8251ACから，1キャラクタ分のデータを引き取ると，$\overline{\mathrm{RDY}}$=0となる．KBは，このような($\overline{\mathrm{RDY}}$=1→$\overline{\mathrm{RDY}}$=0)シーケンスを認識した後に，次の1キャラクタ分のデータを，8251ACに転送して来る．
KBに対するコマンドのKBDE(KB送信ディスエーブル)をセットすることにより，$\overline{\mathrm{RDY}}$=1となり，KBからのデータ送信は禁止される．

④ KBは，$\overline{\mathrm{RDY}}$=0かつ次に示す条件の時，次の動作をする．

| 条 | 件 | 動　　　作 |
|---|---|---|
| $\overline{\mathrm{RDY}}$＝0 | $\overline{\mathrm{RTY}}$＝0 | 直前に転送したシリアルデータと同じデータを，再度転送する． |
| | $\overline{\mathrm{RTY}}$＝1 | 新しいシリアルデータを転送する． |

⑤ キー入力時のリピート機能は，キーボード側が行う．キーが押下されて，Makeデータを発生した後，約0.5秒後までにそのキーが離されない場合，キーボードはそのキーのBreakデータとMakeデータを発生する．その後，キーが離されるまで，60msごとに1度，Break，Makeデータを発生する．60msたたないうちにキーが離された場合，Breakデータのみ発生される．



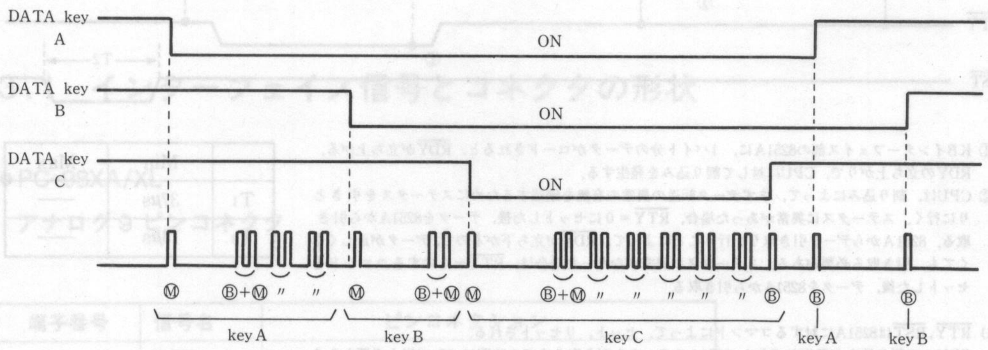

注：対象キーはシフトキー，プログラマブルファンクションキーを除いたすべてのキー

⑥ キー入力方式およびMake，Breakデータ，発生タイミング

キー入力方式は，Nキーロールオーバーとする．

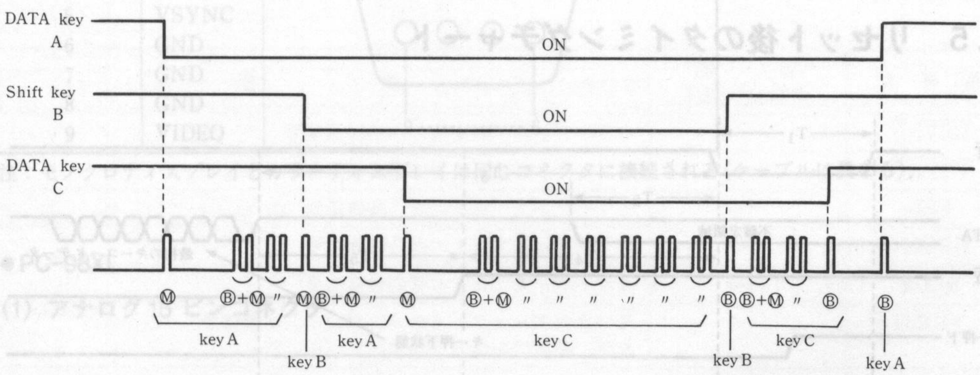
キー押下とMake，Break発生のタイミングは次に示すようになる．

例1




Ⓜ……Makeデータ発生
Ⓑ……Breakデータ発生

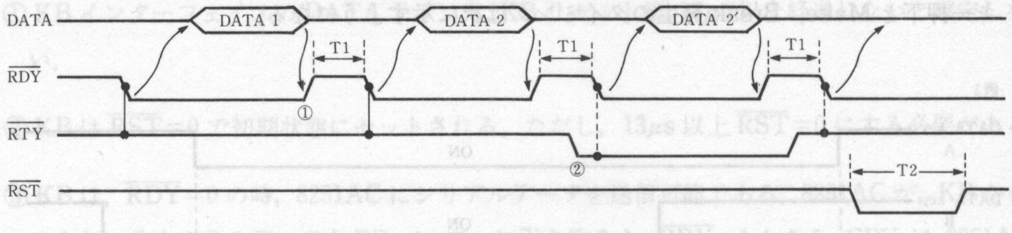
例2




注1：リピートの対象は，より後に押されたKeyに移る．ただし，後に押されたKeyがリピート対象外のKeyの場合は，リピートの対象は移動しない．

注2：リピート対象キーが離されると，新たにリピート対象キーが押されるまで，リピートは発生しない．

## 2.4　転送タイミングチャート



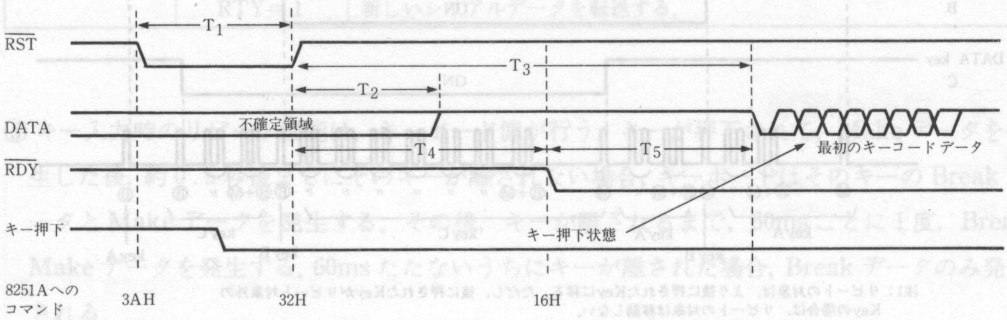

| | Min | Max |
|---|---|---|
| $T_1$ | 37$\mu$s | —— |
| $T_2$ | 13$\mu$s | —— |

① KBインターフェイス部の8251Aに，1バイト分のデータがロードされると，$\overline{RDY}$が立ち上がる．$\overline{RDY}$の立ち上がりで，CPUに対して割り込みを発生する．

② CPUは，割り込みによって，まずデータ転送の異常の有無を確認するためにステータスを引きとりに行く．ステータスに異常があった場合，$\overline{RTY}=0$にセットした後，データを8251Aから引き取る．8251Aからデータ引き取りを行うことによって，$\overline{RDY}$が立ち下がるので，データが正しくなくても，引き取る必要がある．ステータスに異常がなかった場合は，$\overline{RTY}=1$にするコマンドをセットした後，データを8251Aから引き取る．

注）$\overline{RTY}$, $\overline{RST}$は8251Aに対するコマンドによって，セット，リセットされる．
CPUへの割り込みが発生してから，8251Aのデータを引き取るまでの時間は，37$\mu$S以上必要となる（KBに対する$\overline{RDY}=1$のパルス幅は37$\mu$S以上となる）．

## 2.5　リセット後のタイミングチャート




- リセット後$T_2$の間，DATAラインは不確定なので，8251Aに対する受信のイネーブル（RXE＝1）は，リセット後，$T_2$以上経過した後に行うこと．さもないと，8251Aがキーインデータ以外のものを受信する可能性がある．
- キーボードに対する送信イネーブル（KBDE＝0）は，リセット後，$T_2$以上経過した後に行うこと．
- KBDB＝0セット後，最初のデータがKBから送られてくるまでにRXE＝1になっていなければならない．
  8251Aが1度でもデータを取りこぼすと，KBはストールする．したがって，RXE＝1，KBDE＝0セットを同時に行うか，RXE＝1セット後KBDE＝0セットすること．

| | Min | Max |
|---|---|---|
| $T_1$ | 13$\mu$s | —— |
| $T_2$ | —— | 18$\mu$s |

| | $T_3$ | $T_5$ |
|---|---|---|
| $T_2<T_4<12.5$ms の場合 | Min 13ms, Max 16ms | $T_3-T_4$ |
| 12.5ms$\leq T_4$ の場合 | $T_4+T_5$ | Min 100$\mu$s, Max 3.5ms |

# 第3章

# CRTインターフェイス

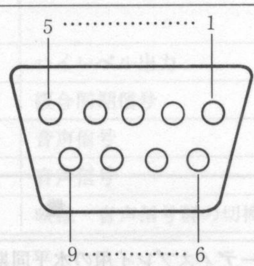
## 3.1 インターフェイス信号とコネクタの形状

●PC-98XA/XL

アナログ9ピンコネクタ

| 端子番号 | 信号名 |
|---|---|
| 1 | RED |
| 2 | GREEN |
| 3 | BLUE |
| 4 | HSYNC |
| 5 | VSYNC |
| 6 | GND |
| 7 | GND |
| 8 | GND |
| 9 | VIDEO |

ピンコネクション



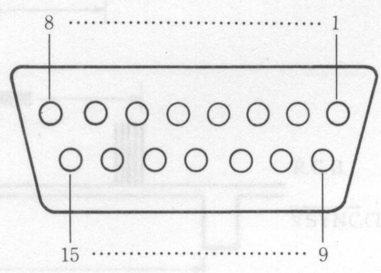

注：モノクロディスプレイとカラーディスプレイは同じコネクタに接続される(ケーブルは異なる).

●PC-98XL

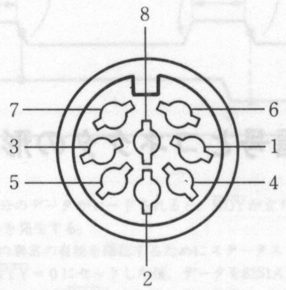
(1) アナログ15ピンコネクタ

| 端子番号 | 信号名 |
|---|---|
| 1 | AR |
| 2 | GND |
| 3 | AG |
| 4 | GND |
| 5 | AB |
| 6 | GND |
| 7 | YS |
| 8 | GND |
| 9 | SYNC |
| 10 | AUDOL |
| 11 | AUDOR |
| 12 | GND |
| 13 | AV |
| 14 | $\overline{\text{HSYNC}}$ |
| 15 | $\overline{\text{VSYNC}}$ |

ピンコネクション



### (2) デジタルRGBコネクタ

| 端子番号 | 信号名 | ピンコネクション |
|---|---|---|
| 1 | +12V |  |
| 2 | GND | |
| 3 | CLOCK | |
| 4 | $\overline{\text{HSYNC}}$ | |
| 5 | $\overline{\text{VSYNC}}$ | |
| 6 | R | |
| 7 | G | |
| 8 | B | |

## 3.2 信号の機能

### ●PC-98XA/XL アナログ9ピン

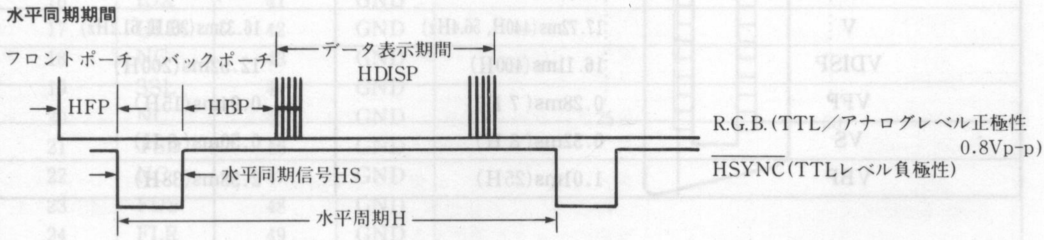
| 信号名 | 方向<br>本体←→CRT | 機能 |
|---|---|---|
| $\overline{\text{HSYNC}}$ | ⟶ | カラーディスプレイ用の水平同期信号 |
| $\overline{\text{VSYNC}}$ | ⟶ | カラーディスプレイ用の垂直同期信号 |
| RED | ⟶ | カラーディスプレイの赤の映像信号 |
| GREEN | ⟶ | カラーディスプレイの緑の映像信号 |
| BLUE | ⟶ | カラーディスプレイの青の映像信号 |
| VIDEO | ⟶ | モノクロディスプレイのコンポジット映像信号<br>水平，垂直同期信号が含まれている |

●PC-98XL アナログ15 ピン／デジタル RGB

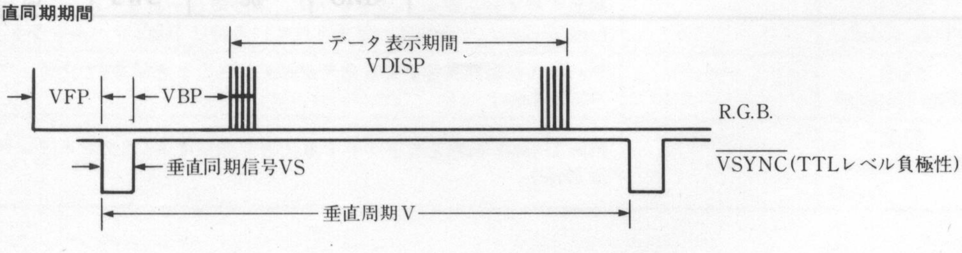
| 信号名 | 方向<br>本体←→CRT | 機能 |
|---|---|---|
| $\overline{\mathrm{HSYNC}}$ | → | カラーディスプレイ用の水平同期信号 |
| $\overline{\mathrm{VSYNC}}$ | → | カラーディスプレイ用の垂直同期信号 |
| R | → | カラーディスプレイの赤の映像信号 |
| G | → | カラーディスプレイの緑の映像信号 |
| B | → | カラーディスプレイの青の映像信号 |
| CLOCK | → | 14.318MHz のクロック出力 |
| VIDEO | → | モノクロディスプレイのコンポジット映像信号<br>水平，垂直同期信号が含まれている |
| LPEN | ← | ライトペンの入力信号 |
| AR | → | アナログ RGB ディスプレイの赤の映像信号 |
| AG | → | 〃　緑　〃 |
| AB | → | 〃　青　〃 |
| YS | → | ハイレベル出力 |
| SYNC | → | 複合同期信号 |
| AUDIOL | → | 音声信号 |
| AUDIOR | → | 音声信号 |
| AV | → | 映像・音声信号群の切換信号 |

## 3.3　タイミングチャート

**水平同期期間**




**垂直同期期間**



●PC-98XA/XLハイレゾリューションモード

| Symbol | タイミング | |
|---|---|---|
| | 奇数フィールド | 偶数フィールド |
| H | 30.45$\mu$s (32.84KHz) | |
| HDISP | 23.41$\mu$s | |
| HFP | 2.34$\mu$s | |
| HS | 1.76$\mu$s | |
| HBP | 2.93$\mu$s | |
| V | 12.5ms (410.5H, 80Hz) | |
| VDISP | 11.42ms (375H) | |
| VFP | 0.244ms (8H) | 0.259ms (8.5H) |
| VS | 0.152ms (5H) | |
| VBP | 0.685ms (22.5H) | 0.670ms (22H) |

●PC-98XLノーマルモード

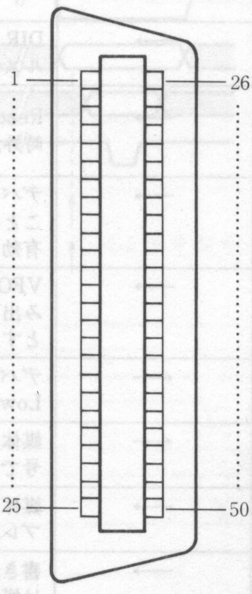
| Symbol　ディスプレイタイプ | 専用高解像度ディスプレイ | 標準ディスプレイ |
|---|---|---|
| H | 40.28$\mu$s(24.83KHz) | 62.58$\mu$s(15.98KHz) |
| HDISP | 30.4$\mu$s | 44.70$\mu$s |
| HFP | 3.04$\mu$s | 4.47$\mu$s |
| HS | 3.04$\mu$s | 4.47$\mu$s |
| HBP | 3.8$\mu$s | 8.94$\mu$s |
| V | 17.72ms(440H, 56.4Hz) | 16.33ms(261H, 61.2Hz) |
| VDISP | 16.11ms(400H) | 12.52ms(200H) |
| VFP | 0.28ms(7H) | 0.94ms(15H) |
| VS | 0.32ms(8H) | 0.50ms(8H) |
| VBP | 1.01ms(25H) | 2.38ms(38H) |

# 第4章

# フロッピーディスクインターフェイス

## 4.1 1MBFD インターフェイス

### 4.1.1 インターフェイス信号とコネクタの形状

| 端子番号 | 信号名 | 端子番号 | 信号名 | ピンコネクション |
|---|---|---|---|---|
| 1 | WID | 26 | GND |  |
| 2 | MFM | 27 | GND | |
| 3 | RDT | 28 | GND | |
| 4 | PRT | 29 | GND | |
| 5 | TK0 | 30 | GND | |
| 6 | WGT | 31 | GND | |
| 7 | WDT | 32 | GND | |
| 8 | STP | 33 | GND | |
| 9 | DIR | 34 | GND | |
| 10 | DS4 | 35 | GND | |
| 11 | DS3 | 36 | GND | |
| 12 | DS2 | 37 | GND | |
| 13 | DS1 | 38 | GND | |
| 14 | SYC | 39 | GND | |
| 15 | RDY | 40 | GND | |
| 16 | IDX | 41 | GND | |
| 17 | HLD | 42 | GND | |
| 18 | NC | 43 | GND | |
| 19 | SSL | 44 | GND | |
| 20 | NC | 45 | GND | |
| 21 | TSD | 46 | GND | |
| 22 | NC | 47 | GND | |
| 23 | FUS | 48 | GND | |
| 24 | FLR | 49 | GND | |
| 25 | LWC | 50 | GND | |

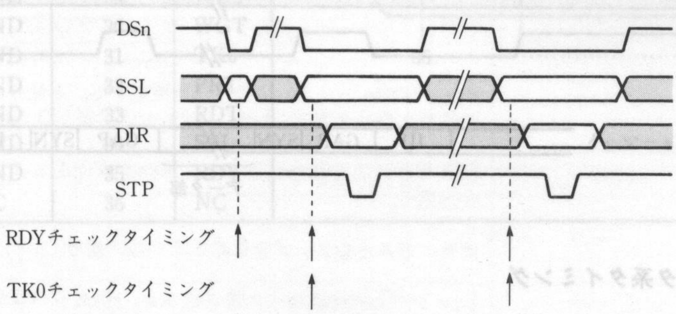
## 4.1.2 信号の機能

| 信号名 | 方向 本体←→FD | 機能 |
|---|---|---|
| WID (Window) | ← | VFO回路から出力されるデータとクロックの弁別を行う．本信号とRDT信号の位相合わせは μPD765A で行う． |
| MFM (MFM Mode) | → | MFM記録方式のデータの読み出し，書き込みを行うことを指定する(アクティブレベルはLow)． |
| RDT (Read Data) | ← | 媒体から読み出したデータ信号．VFO回路によってWID信号と同期した信号でなければならない． |
| PRT (Write Protect) | ← | 媒体への書き込みが禁止状態であることを示す(アクティブレベルはLow)． |
| TK0 (Track 00) | ← | ヘッドが00トラック上にあることを示す． |
| WGT (Write Gate) | → | Lowレベルの時媒体への書き込みを，Highレベルの時媒体からの読み出しを指示する． |
| WDT (Write Data) | → | 媒体に書き込むデータを供給するパルス信号． |
| STP (Step) | → | DIR信号により指定した方向へRead/Writeヘッドを移動させるパルス信号． |
| DIR (Direction Select) | → | Read/Writeヘッドの移動方向を指定する．本信号がHighレベルの時外周トラック方向，Lowレベルの時内周トラック方向を指定する． |
| DS1～4 (Drive Select 1～4) | → | デバイスを選択する．DS1～4の内いずれか1つをLowレベルにすることにより，対応するデバイスが選択され，その他の入出力信号線が有効となる． |
| SYC (VFO Sync) | → | VFOの動作モードを指定する．本記号がLowレベルの時，媒体の読み出しデータへの同期動作を指示する．非読み出し時はHighレベルとする． |
| RDY (Ready) | ← | デバイスが動作可能状態であることを示す(アクティブレベルはLow)． |
| IDX (Index) | ← | 媒体上の起点を示す．媒体が1回転するごとに1回出力するパルス信号である． |
| HLD (Head Load) | → | 媒体面にRead/Writeヘッドをロードすることを指定する(アクティブレベルはLow)． |
| SSL (Side Select) | → | 書き込み／読み出しに使用する媒体面を選択する．Highレベルの時は媒体の〝0″面側のヘッドを選択し，Lowレベルの時は媒体の〝1″面側のヘッドを選択する． |
| TSD (Two Sided) | ← | 装着されている媒体の種類を示す．両面媒体が装着されている時はLowレベル，片面媒体が装着されている時はHighレベルとなる． |
| FUS (File Unsafe) | ← | デバイスが動作異常となる条件が発生したことを示す(アクティブレベルはLow)． |
| FLR (File Unsafe Reset) | → | File Unsafe状態をリセットすることを指定する(アクティブレベルはLow)． |

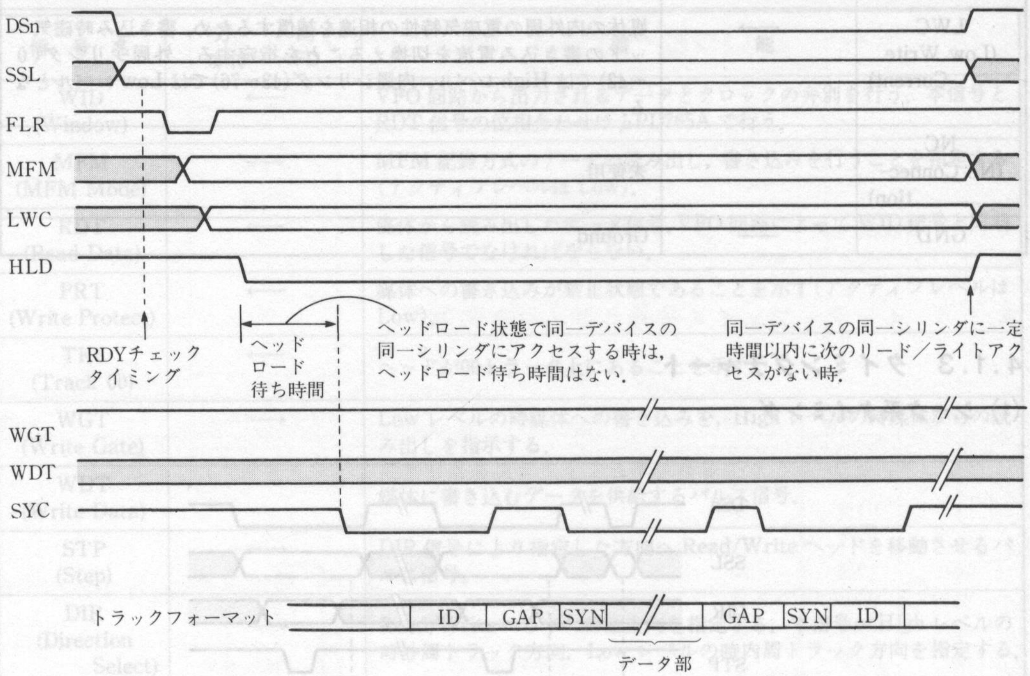
| 信号名 | 方向<br>本体←→FD | 機能 |
|---|---|---|
| LWC (Low Write Current) | ──→ | 媒体の内外周の電磁気特性の相違を補償するため，書き込み時磁気ヘッドの書き込み電流を切換えることを指定する．外周シリンダ(0～42)ではHighレベル，内周シリンダ(43～76)ではLowレベルとなる． |
| NC (No Connection) | | 未使用 |
| GND | ── | Ground |

## 4.1.3　タイミングチャート

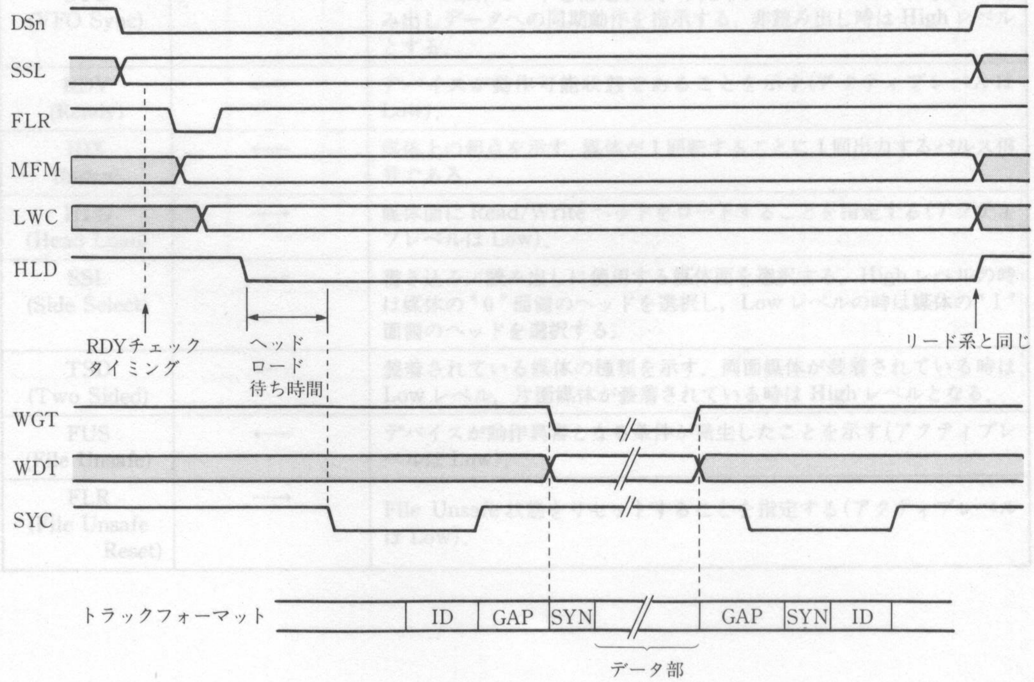
### (1) シーク系タイミング



## (2) リードデータ系タイミング

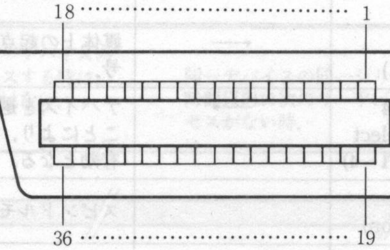



## (3) ライトデータ系タイミング



# 4.2 640KBFD インターフェイス

## 4.2.1 インターフェイス信号とコネクタの形状

| 端子番号 | 信号名 | 端子番号 | 信号名 | ピンコネクション |
|---|---|---|---|---|
| 1 | GND | 19 | NC |  |
| 2 | GND | 20 | HLD | |
| 3 | GND | 21 | DS4 | |
| 4 | GND | 22 | IDX | |
| 5 | GND | 23 | DS1 | |
| 6 | GND | 24 | DS2 | |
| 7 | GND | 25 | DS3 | |
| 8 | GND | 26 | MTR | |
| 9 | GND | 27 | DIR | |
| 10 | GND | 28 | STP | |
| 11 | GND | 29 | WDT | |
| 12 | GND | 30 | WGT | |
| 13 | GND | 31 | TK0 | |
| 14 | GND | 32 | PRT | |
| 15 | GND | 33 | RDT | |
| 16 | GND | 34 | SSL | |
| 17 | GND | 35 | RDY | |
| 18 | NC | 36 | NC | |

## 4.2.2 信号の機能

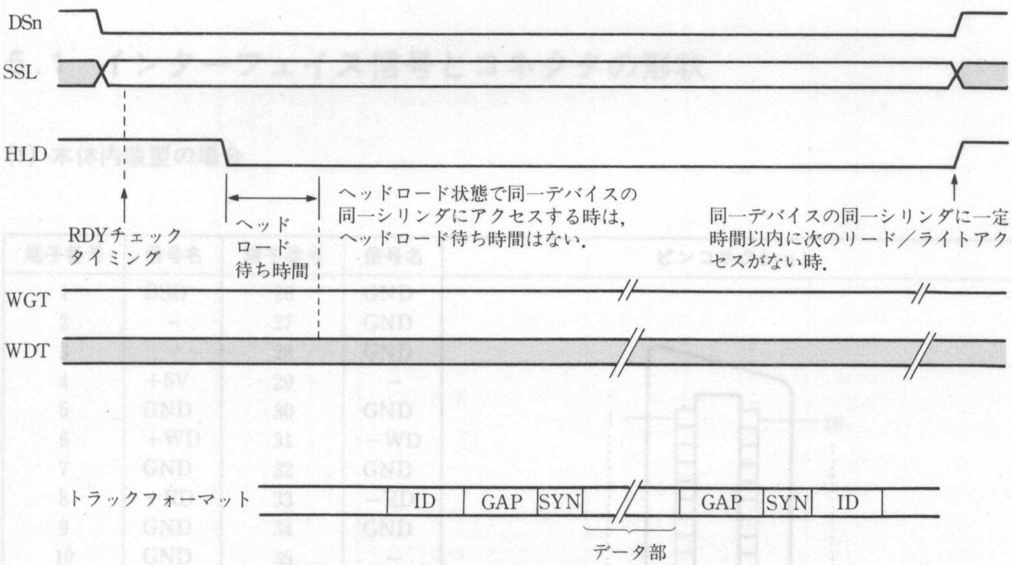
| 信号名 | 方向 本体←→FD | 機能 |
|---|---|---|
| GND | ―― | Ground |
| NC (No Connection) | | 未使用 |
| HLD (Head Load) | → | 媒体面にRead/Writeヘッドをロードすることを指定する(アクティブレベルはLow). |
| IDX (Index) | ← | 媒体上の起点を示す．媒体が1回転するごとに1回出力するパルス信号. |
| DS1～4 (Drive Select 1～4) | → | デバイスを選択する．DS1～4の内いずれか1つをLowレベルにすることにより，対応するデバイスが選択され，その他の入出力信号線が有効となる. |
| MTR (Motor on) | → | スピンドルモータを回転させる(アクティブレベルはLow). |
| DIR (Direction Select) | → | Read/Writeヘッドの移動方向を指定する．本信号がHighレベルの時外周トラック方向，Lowレベルの時内周トラック方向を指定する. |
| STP (Step) | → | DIR信号により指定した方向へRead/Writeヘッドを移動させるパルス信号. |
| WDT (Write Data) | → | 媒体に書き込むデータを供給するパルス信号. |
| WGT (Write Gate) | → | Lowレベルの時媒体への書き込みを，Highレベルの時媒体からの読み出しを指示する. |
| TK0 (Track 00) | ← | ヘッドが00トラック上にあることを示す．(アクティブレベルはLow). |
| PRT (Write Protect) | ← | 媒体への書き込みが禁止状態であることを示す(アクティブレベルはLow). |
| RDT (Read Data) | ← | 媒体から読出したデータ信号．VFO回路によってWID信号と同期した信号でなければならない. |
| SSL (Side Select) | → | 書き込み／読み出しに使用する媒体面を選択する．Highレベルの時は媒体の〝0〞面側のヘッドを選択し，Lowレベルの時は媒体の〝1〞面側のヘッドを選択する. |
| RDY (Ready) | ← | デバイスが動作可能状態であることを示す(アクティブレベルはLow). |

### 4.2.3　タイミングチャート

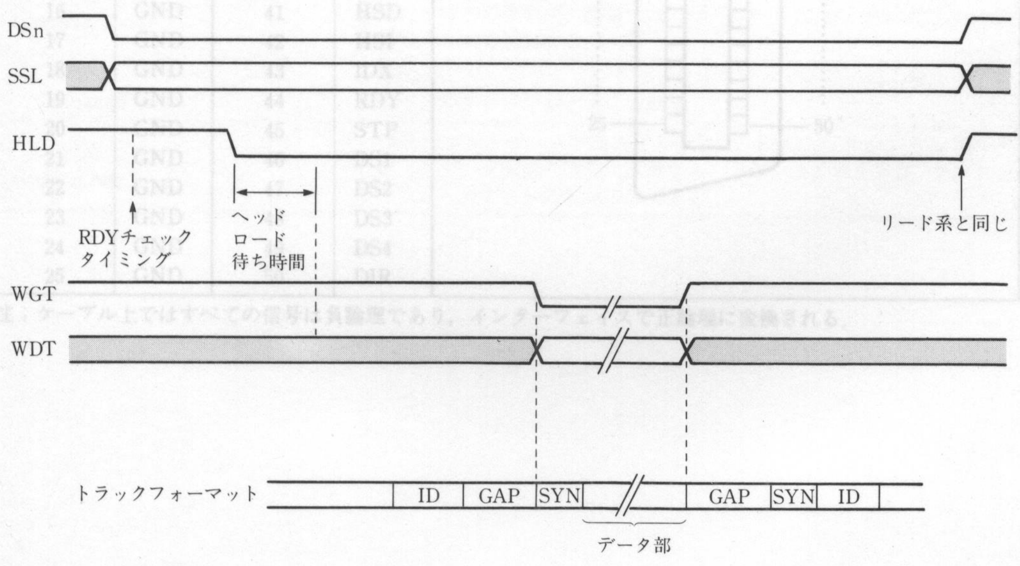
**(1) シーク系タイミング**

1MBFDの場合と同じ

**(2) リードデータ系タイミング**




**(3) ライトデータ系タイミング**



## 4.2.2 信号の機能

| 信号名 | 方向 本体←→FD | 機能 |
|---|---|---|
| GND | —— | Ground |
| NC (No Connection) | | 未使用 |
| HLD (Head Load) | → | 媒体面にRead/Writeヘッドをロードすることを指定する(アクティブレベルはLow)。 |
| IDX (Index) | ← | 媒体上の基点を示す。媒体が1回転するごとに1回出力するパルス信号。 |
| DS1～4 (Drive Select 1～4) | → | デバイスを選択する。DS1～4のいずれか1つをLowレベルにすることにより、対応するデバイスが選択され、その他の入出力信号線が有効となる。 |
| MTR (Motor On) | → | スピンドルモータを回転させる(アクティブレベルはLow)。 |
| DIR (Direction Select) | → | Read/Writeヘッドの移動方向を指定する。本信号がHighレベルの時外周トラック方向、Lowレベルの時内周トラック方向を指定する。 |
| STP (Step) | → | DIR信号により指定した方向へRead/Writeヘッドを移動させるパルス信号。 |
| WDT (Write Data) | → | 媒体に書き込むデータを供給するパルス信号。 |
| WGT (Write Gate) | → | Lowレベルの時媒体への書き込みを、Highレベルの時媒体からの読み出しを指示する。 |
| TK0 (Track 00) | ← | ヘッドが00トラック上にあることを示す。(アクティブレベルはLow)。 |
| PRT (Write Protect) | ← | 媒体への書き込みが禁止状態であることを示す(アクティブレベルはLow)。 |
| RDT (Read Data) | ← | 媒体から読出したデータ信号。VFO回路によってWID信号と同期した信号でなければならない。 |
| SSL (Side Select) | → | 書き込み／読み出しに使用する媒体面を選択する。Highレベルの時は媒体の"0"面側のヘッドを選択し、Lowレベルの時は媒体の"1"面側のヘッドを選択する。 |
| RDY (Ready) | ← | デバイスが動作可能状態であることを示す(アクティブレベルはLow)。 |

# 第5章

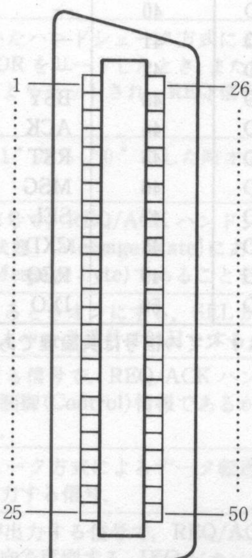
# 固定ディスクインターフェイス

## 5.1　インターフェイス信号とコネクタの形状

### (1) 本体内蔵型の場合

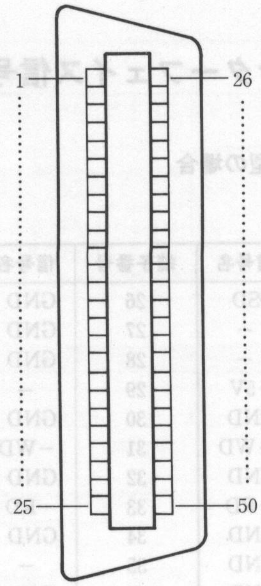
| 端子番号 | 信号名 | 端子番号 | 信号名 |
|---|---|---|---|
| 1 | DSD | 26 | GND |
| 2 | − | 27 | GND |
| 3 | | 28 | GND |
| 4 | +5V | 29 | − |
| 5 | GND | 30 | GND |
| 6 | +WD | 31 | −WD |
| 7 | GND | 32 | GND |
| 8 | +RD | 33 | −RD |
| 9 | GND | 34 | GND |
| 10 | GND | 35 | − |
| 11 | GND | 36 | HS2 |
| 12 | GND | 37 | WGT |
| 13 | GND | 38 | SKC |
| 14 | GND | 39 | TRO |
| 15 | GND | 40 | FLT |
| 16 | GND | 41 | HSD |
| 17 | GND | 42 | HSI |
| 18 | GND | 43 | IDX |
| 19 | GND | 44 | RDY |
| 20 | GND | 45 | STP |
| 21 | GND | 46 | DS1 |
| 22 | GND | 47 | DS2 |
| 23 | GND | 48 | DS3 |
| 24 | GND | 49 | DS4 |
| 25 | GND | 50 | DIR |

ピンコネクション




注：ケーブル上ではすべての信号は負論理であり，インターフェイスで正論理に変換される．

## (2) PC-9801-27 固定ディスクインターフェイスボード(H/A バス)

| 端子番号 | 信号名 | 端子番号 | 信号名 | ピンコネクション |
|---|---|---|---|---|
| 1 | GND | 26 | DT0 |  |
| 2 | GND | 27 | DT1 | |
| 3 | GND | 28 | DT2 | |
| 4 | GND | 29 | DT3 | |
| 5 | GND | 30 | DT4 | |
| 6 | GND | 31 | DT5 | |
| 7 | GND | 32 | DT6 | |
| 8 | GND | 33 | DT7 | |
| 9 | GND | 34 | — | |
| 10 | GND | 35 | — | |
| 11 | GND | 36 | — | |
| 12 | GND | 37 | — | |
| 13 | GND | 38 | — | |
| 14 | GND | 39 | — | |
| 15 | GND | 40 | — | |
| 16 | GND | 41 | — | |
| 17 | GND | 42 | — | |
| 18 | GND | 43 | BSY | |
| 19 | GND | 44 | ACK | |
| 20 | GND | 45 | RST | |
| 21 | GND | 46 | MSG | |
| 22 | GND | 47 | SEL | |
| 23 | GND | 48 | CXD | |
| 24 | GND | 49 | REQ | |
| 25 | GND | 50 | IXO | |

注：ケーブル上ではすべての信号は負論理であり，インターフェイスボードで正論理に変換される．

## 5.2　信号の機能

固定ディスクユニットのインターフェイス信号をH/Aバス信号と呼ぶ．H/Aバス信号は，データ信号と制御信号に大別される(固定ディスクを本体内に内蔵するモデルの場合，H/Aバスは外部には出ていない)．

1) データ信号は，8本のデータ信号で構成され，バイト単位のデータが転送される．
　データ転送の制御は後述のREQ/ACK信号を用いた，ハンドシェーク方式による．

2) 制御信号は次に示す8本がある．

| 信号名 | 方向<br>本体←→HD | 機能 |
|---|---|---|
| BSY<br>(Busy) | ← | H／Aバスが動作中であることを示す． |
| ACK<br>(Acknowledge) | → | REQ/ACK信号を用いたハンドシェーク方式によるデータ転送の際の応答信号．ACKはIDRをリードしたとき，またはODRにライトしたときハードウェアによりセットされ，REQ信号がオフになるとリセットされる． |
| RST<br>(Reset) | → | CCRのRST bitを〝1〃から〝0〃にした時オンになり，コントローラをセットする． |
| MSG<br>(Message) | ← | コントローラの出力信号で，REQ/ACKハンドシェークによるデータ転送が，「動作完了状態」(Message State)にあり，情報が「ポストステータスバイト」(Message byte)であることを示す． |
| SEL<br>(Select) | → | コントローラを選択するときオンにする．SELがオン時のデータ信号は，選択すべきコントローラ番号で01Hとする． |
| CXD<br>(Control/Data) | ← | コントローラが出力する信号で，REQ/ACKハンドシェークによるデータ転送の情報が，制御(Control)情報であるか，データ(Data)情報であるかを区別する． |
| REQ<br>(Request) | ← | REQ/ACKハンドシェーク方式によるデータ転送の際の要求信号としてコントローラが出力する信号． |
| IXO<br>(Input/Output) | ← | IXOはコントローラが出力する信号で，REQ/ACKハンドシェークによるデータ転送の方向を区別する．IXOがオンのときは，コントローラからH／Aへの転送方向である． |

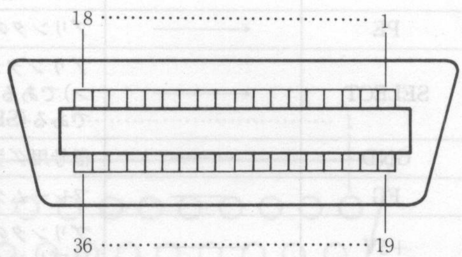
# 第6章

# プリンタインターフェイス

## 6.1 インターフェイス信号とコネクタの形状

| 端子番号 | 信号名 | 端子番号 | 信号名 |
|---|---|---|---|
| 1 | DATA·STB | 19 | GND |
| 2 | RDB0 | 20 | GND |
| 3 | RDB1 | 21 | GND |
| 4 | RDB2 | 22 | GND |
| 5 | RDB3 | 23 | GND |
| 6 | RDB4 | 24 | GND |
| 7 | RDB5 | 25 | GND |
| 8 | RDB6 | 26 | GND |
| 9 | RDB7 | 27 | GND |
| 10 | $\overline{\text{ACK}}$ | 28 | GND |
| 11 | BUSY | 29 | GND |
| 12 | PE | 30 | GND |
| 13 | SELECT | 31 | $\overline{\text{INP·PRIME}}$ |
| 14 | GND | 32 | FAULT |
| 15 | NC | 33 | GND |
| 16 | GND | 34 | NC |
| 17 | FG | 35 | NC |
| 18 | +5V | 36 | $\overline{\text{INP·BUSY}}$ |

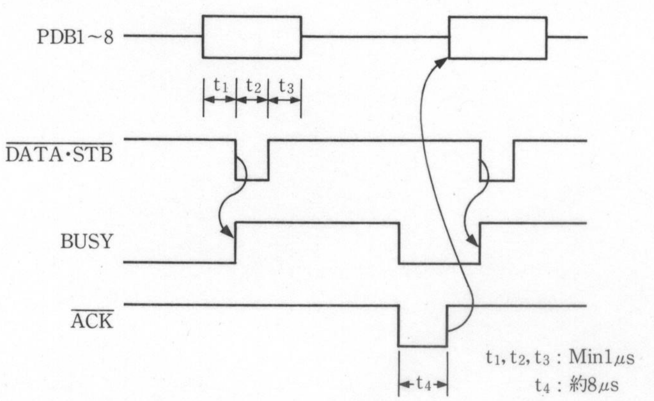
ピンコネクション



## 6.2　信号の機能

| 信号名 | 方向<br>本体↔プリンタ | 機能 |
|---|---|---|
| $\overline{\mathrm{DATA \cdot STB}}$ | ⟶ | データをプリンタに伝えるための同期信号.<br>定常状態はHIGHである. プリンタはこの信号がHIGHからLOWになった時BUSYをHIGHにし, LOWからHIGHになった時, データを読み込む. パルス幅は1μs以上である. |
| PDB1<br>~PDB8 | ⟶ | 8ビットのデータを送信する出力信号.<br><br>論理1はHIGHである. PDB1がLSB, PDB8がMSBとなる. パルス幅は3μs以上である. |
| $\overline{\mathrm{ACK}}$ | ⟵ | プリンタが, 装置内へのデータの取り込みを完了したことを示す信号. $\overline{\mathrm{DATA \cdot STB}}$に対する応答である. ただし, プリンタからは電源投入時に無条件に一度出力されることがある. |
| BUSY | ⟵ | プリンタがデータ受信不可能(BUSY)状態であることを示す信号. LOWの場合, プリンタはデータ受信が可能である. |
| PE | ⟵ | プリンタの用紙が残り少なくなった時HIGHになる. |
| SELECT | ⟵ | プリンタがSELECT(オンライン)であるかDESELECT(オフライン)であるかを示す信号. HIGHの場合, プリンタはデータ受信が可能である(SELECT). |
| GND | —— | 信号用グランド |
| FG | —— | フレームグランド |
| +5V | ⟵ | プリンタの電源部からの信号. プリンタの電源ON/OFFの状態. またはプリンタケーブル接続の状態が確認できる. |
| $\overline{\mathrm{INPUT \cdot}}$<br>$\overline{\mathrm{PRIME}}$ | ⟶ | プリンタを初期化するための信号. パルス幅は15μs以上. 一定時間以上LOWにすると, バッファ内のデータを処理した後, プリンタは初期状態になる(SELECT, DESELECTの状態のみは変化しない). |
| $\overline{\mathrm{FAULT}}$ | ⟵ | プリンタがエラーを起したことを示す信号<br>(エラー以外に, PE＝1, またはSELECT＝0の場合にもLOWとなる). |
| INP・BUSY | ⟵ | 通常BUSY信号と同一である. ただし, プリンタによっては, データを受信してから装置内への取り込みが完了するまでの間はBUSY信号はLOWのままで, この信号がHIGHになるものがある. PC-PR201系統のプリンタにはこの信号はない. |

## 6.3　タイミングチャート



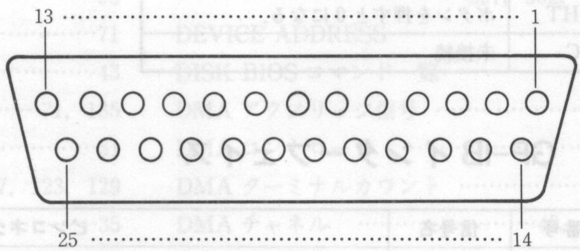
# 第7章

# その他のインターフェイス

## 7.1 RS-232C インターフェイス

| 端子番号 | 信号名 |
|---|---|
| 1 | GND |
| 2 | TXD |
| 3 | RXD |
| 4 | RTS |
| 5 | CTS |
| 6 | DSR |
| 7 | GND |
| 8 | DCD |
| 9 | NC |
| 10 | NC |
| 11 | GND |
| 12 | NC |
| 13 | GND |
| 14 | GND |
| 15 | TXC (2) |
| 16 | NC |
| 17 | RXC |
| 18 | NC |
| 19 | NC |
| 20 | DTR |
| 21 | NC |
| 22 | RI |
| 23 | NC |
| 24 | TXC (1) |
| 25 | NC |

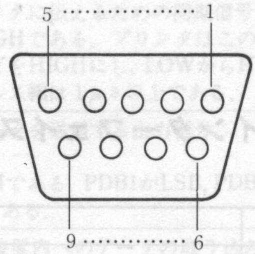
ピンコネクション




(1)送信エレメントタイミング 1　(2)送信エレメントタイミング 2

## 7.2　マウスインターフェイス

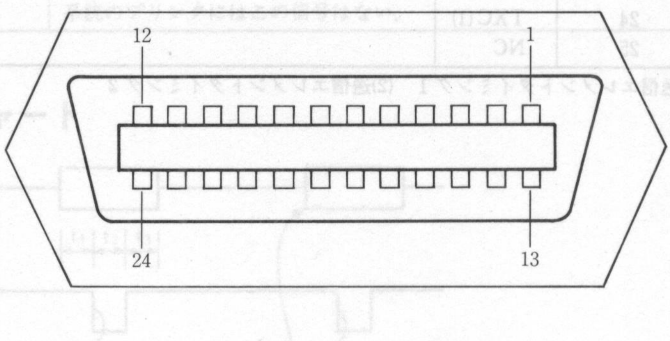
| 端子番号 | 信号名 | ピンコネクション |
|---|---|---|
| 1 | +5V | |
| 2 | XA | |
| 3 | XB | |
| 4 | YA | |
| 5 | YB | |
| 6 | LEFT | |
| 7 | NC | |
| 8 | RIGHT | |
| 9 | GND | |




| 信号名 | 説　　　明 |
|---|---|
| XA, XB<br>YA, YB | マウスからのエンコーダ入力 |
| LEFT<br>RIGHT | マウス上にあるボタンの入力<br>ボタンを押すと0になる. |
| NC | 未接続 |

## 7.3　GP-IBインターフェイス

| 端子番号 | 信号名 | ピンコネクション |
|---|---|---|
| 1 | DIO1 | |
| 2 | DIO2 | |
| 3 | DIO3 | |
| 4 | DIO4 | |
| 5 | EOI | |
| 6 | DAV | |
| 7 | NRFD | |
| 8 | NDAC | |
| 9 | IFC | |
| 10 | SRQ | |
| 11 | ATN | |
| 12 | シールド | |
| 13 | DIO5 | |
| 14 | DIO6 | |
| 15 | DIO7 | |
| 16 | DIO8 | |
| 17 | REN | |
| 18 | GND | |
| 19 | GND | |
| 20 | GND | |
| 21 | GND | |
| 22 | GND | |
| 23 | GND | |
| 24 | ロジック<br>GND | |



# 索　引

## タ

PC-98XL/XA
テクニカルデータブック

1987年5月1日　初版発行

定　価　6,000円

編　集　アスキー出版局テクライト
発行者　塚本慶一郎
発行所　株式会社アスキー
〒107　東京都港区南青山6-11-1　スリーエフ南青山ビル
振　替　東京4-161144
電　話　03-486-7111(代表)
情報電話　03-498-0299(ダイヤルイン)
出版営業部　03-486-1977(ダイヤルイン)



編集担当　永島智二
制作担当　松村礼司
表紙担当　古屋佳子
プログラム協力　赤嶋映子
C T S　福田工芸株式会社
制　作　株式会社ガロ
印　刷　壮光舎印刷株式会社

ISBN4-87148-424-6 C3055 ¥6000E

■対応機種

- ●PC-98XA model 1
- ●PC-98XA model 2
- ●PC-98XA model 3
- ●PC-98XA model 11
- ●PC-98XA model 21
- ●PC-98XA model 31
- ●PC-98XL model 1
- ●PC-98XL model 2
- ●PC-98XL model 4

ISBN4-87148-424-6 C3055 ¥6000E

定価6,000円